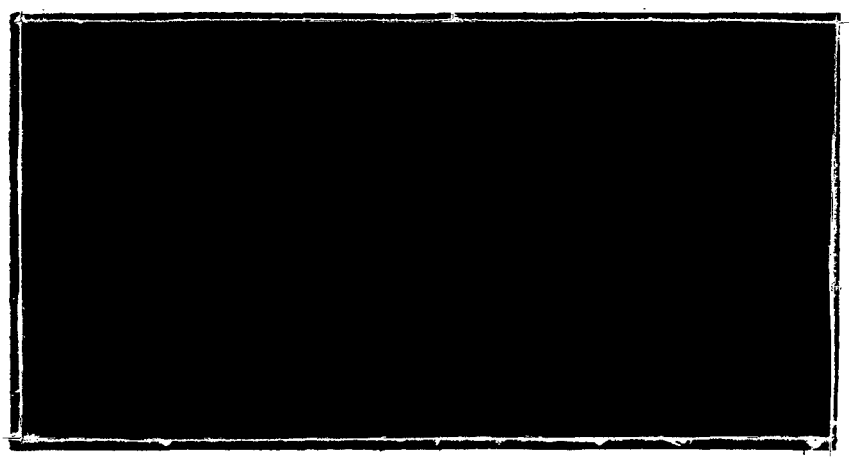


DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/DS/ENG/97-02

PHYSIOLOGICALLY-BASED VISION MODELING APPLICATIONS AND GRADIENT DESCENT-BASED PARAMETER ADAPTATION OF PULSE COUPLED NEURAL NETWORKS

DISSERTATION
Randy Paul Broussard
Captain, USAF

AFIT/DS/ENG/97-02

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

AFIT/DS/ENG/97-02

PHYSIOLOGICALLY-BASED VISION MODELING APPLICATIONS AND GRADIENT DESCENT-BASED PARAMETER ADAPTATION OF PULSE COUPLED NEURAL NETWORKS

DISSERTATION

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Randy Paul Broussard, B.S.E.E., M.S.C.P.

Captain, USAF

June 1997

AFIT/DS/ENG/97-02

PHYSIOLOGICALLY-BASED VISION MODELING APPLICATIONS AND GRADIENT DESCENT-BASED PARAMETER ADAPTATION OF PULSE COUPLED NEURAL NETWORKS

Randy Paul Broussard, B.S.E.E., M.S.C.P.

Captain, USAF

Approved:

Steven K. Rogers Chairman, Advisory Committee	2 JUN 97 Date
Martin P. DeSimio Member, Advisory Committee	22 May 97 Date
Matthew Kabrisky Member, Advisory Committee	22 May 97 Date
Mark E. Oxley Member, Advisory Committee	22 May 97 Date
James T. Moore Dean's Representative	2 Jun 97 Date

Accepted:

Robert A. Calico, Jr.
Dean, Graduate School of Engineering

Acknowledgements

This effort is dedicated to my loving wife Susan. I thank her for the love, patience, and understanding she provided during this work filled process. With her support, I've finally made this dream a reality.

I wish to thank my advisor and committee chairman, Dr. Steven Rogers, for his tireless and unfailing support. His clear and ever humorous guidance kept my path clear and unwavering. I would also like to thank my committee members: Dr. Martin Desimio, Dr. Mark Oxley, and Dr. Mathew Kabrisky whose insight added greatly to the success of this research. I give special thanks to Dr. John Johnson whose tutoring gave me a running start on this new and unfamiliar topic. I thank Dr. (Lt Col) James Moore for his meticulous review which produced a much improved document. Additional thanks goes to Dan Zambon and Dave Doak, whose computer support insured my productivity even when my research brought the network to a crawl.

Randy Paul Broussard

Table of Contents

List of Figures

List of Tables

Abstract

In this research, pulse coupled neural networks (PCNNs) are analyzed and evaluated for use in primate vision modeling. An adaptive PCNN is developed that automatically sets near-optimal parameter values to achieve a desired output. For vision modeling, a physiologically motivated vision model is developed from current theoretical and experimental biological data. The biological vision processing principles used in this model, such as spatial frequency filtering, competitive feature selection, multiple processing paths, and state dependent modulation are analyzed and implemented to create a PCNN based feature extraction network. This network extracts luminance, orientation, pitch, wavelength, and motion, and can be cascaded to extract texture, acceleration and other higher order visual features. Theorized and experimentally confirmed cortical information linking schemes, such as state dependent modulation and temporal synchronization are used to develop a PCNN-based visual information fusion network. The network is used to fuse the results of several object detection systems for the purpose of enhanced object detection accuracy. On actual mammograms and FLIR images, the network achieves an accuracy superior to any of the individual object detection systems it fused. Last, this research develops the first fully adaptive PCNN. Given only an input and a desired output, the adaptive PCNN will find all parameter values necessary to approximate that desired output. A simplified, mathematically equivalent, persistent signal PCNN neuron model is developed and gradient descent is applied to derive parameter adaptation equations (training rules) for all parameters. Implementing these equations forms a fully adaptive PCNN that minimizes squared error between the actual and desired output. All equations can be applied after PCNN execution is complete allowing adaptation to be added to an existing PCNN without any internal modifications.

PHYSIOLOGICALLY-BASED VISION MODELING APPLICATIONS AND GRADIENT DESCENT-BASED PARAMETER ADAPTATION OF PULSE COUPLED NEURAL NETWORKS

I. Introduction

1.1 Historical Background

Computer vision is a large and growing area of research within both the civilian and military communities. Advances in computer vision would allow many tasks to be performed with a quality and precision currently unachievable. One area of computer vision that is still in its infancy is object detection. Three fundamental questions that arise in object detection research are: 1) which characteristics do we extract, 2) how do we extract these characteristics, and 3) how do we combine the features for use in a decision making process? In an attempt to answer these questions, this research examines the methods the biological vision system uses for object detection.

The biological vision system is the best general object detection/recognition system known to exist. Solutions to many of the problems man wishes to solve currently exist in nature. Nature has already found a solution to the problem of object detection. The biological vision system can perform object detection feats that are beyond the capability of current technology. Our vision system filters unwanted information and combines features in a way that allows us to detect and identify objects in our surroundings. It combines many types of visual information to construct our view of the external world. Size, form, motion, color, and texture are identified and combined in a way which allows us to detect and recognize objects. Understanding and simulating the methods the biological vision system uses to extract, select, and combine visual features for object detection is one focus of this research.

In an attempt to discover the secrets of the biological vision system, many physiologically motivated object detection/recognition models have been designed and applied with varying success (25, 12, 35, 36, 37, 38, 39, 40, 60, 66). A tool often used in these object detection/recognition systems is the artificial neural network (77, 66, 86, 57, 67, 66, 12, 11, 28, 79, 18, 74, 1, 17, 78). These networks are biologically inspired combinations of artificial neurons used to simulate theorized neuronal processing. Several neural networks, such as the multi-layer perceptron (back propagation network), adaptive resonance theory network (ART), and Hopfield network, have dominated the vision system modeling area of research (77, 93, 80).

A new neural network called the pulse coupled neural network (PCNN) has shown great promise in the areas of image processing, scene segmentation, pattern recognition, auditory recognition, object time signatures and syntactical computing (84, 76, 22, 56, 83, 54, 52, 49, 43, 55, 5, 53, 50, 73, 95, 4, 51, 42, 41). The PCNN contains several unique physiologically motivated features not contained in the mainstream neural networks (23, 46, 45, 47, 74). The PCNN models the physiologically motivated phenomenon of temporal synchronization which is theorized as the method used to link related information within the brain. It is theorized that biological neurons synchronize and pulse at the same frequency to represent objects or pieces of objects in the visual system (32, 23, 67, 92). The PCNN implements this pulse level synchronization through a physiologically motivated modulatory mechanism. This mechanism can also be used to model other biologically observed phenomenon such as state dependent modulation which can be used in feature extraction (64). These unique features make the PCNN highly suitable for modeling processes in the biological vision system.

A drawback of using the PCNN is the large number of parameters whose values that must be determined. In its simplest form, the PCNN contains 25 adjustable parameters. Many parameters are dependent upon other parameters which makes achieving a desired output difficult (6). To date, guidance on setting PCNN parameters is almost non-existent and no PCNN currently exists

that can automatically adapt parameter values to achieve a desired output. This research extends the state-of-art in PCNN technology by developing the first adaptive PCNN. Given only an input and a desired output, the adaptive PCNN algorithm will find all parameter values necessary to approximate the desired output.

1.2 Problem Statement and Scope

This research 1) demonstrates the usefulness of the PCNN for modeling observed biological *feature extraction*, 2) demonstrates the usefulness of the PCNN for modeling theorized biological *information fusion*, and 3) develops the first PCNN that can automatically adapt parameter values to achieve a desired output.

Current knowledge of the primate vision system is examined for methods that can be used to advance the feature extraction and information fusion portion of the computer vision quest. Current theoretical and experimental data are used to model biological vision processes using the PCNN. The usefulness of this vision model for feature extraction, information fusion, and object detection is demonstrated on the real-world problems of breast cancer detection and SCUD missile launcher detection. Synopses of the three focus areas of this research are presented below.

Physiologically motivated feature extraction using the PCNN and Gabor filters. Feature extraction is modeled using the biologically observed vision processes of spatial frequency filtering (3, 20, 30, 62, 72, 88), competitive feature selection (38, 60, 25, 66, 40, 39, 12, 36, 35, 37), and state dependent modulation (23, 64). Mechanisms inherent to the PCNN are used to implement these feature extraction principles to form a physiologically motivated feature extraction network. To remove unwanted visual information and focus on desired objects, these same vision principles are used to implement a focus of attention mechanism within the network. Features such as (but not limited to) orientation, pitch, intensity, wavelength, and motion at each location in a visual scene

can be extracted with this network. Feature extraction of a subset of these features is demonstrated on gray-scale images.

Physiologically motivated information fusion for object detection using the PCNN. Theorized and experimentally confirmed cortical information linking schemes, such as state dependent modulation and temporal synchronization (32, 23, 64) are used as possible methods of visual information fusion. The PCNN is used to implement these physiologically motivated information linking methods to form a physiologically motivated information fusion network. Using the features and focus of attention provided by the feature extraction network, the information fusion network performs object detection. On two sets of images, the information fusion network produces a reduced false alarm rate compared to two published object detection techniques.

An adaptive PCNN. Gradient descent-based backward error propagation is used to derive parameter adaptation equations (training rules) for all PCNN parameters. Through an analysis of the PCNN neuron, connectivity, and pulse coupling mechanism, adaptation equations are derived for the purpose of automatically adjusting all parameters to approximate a desired output. Implementing these equations forms a fully adaptive PCNN which automatically adapts parameter values to minimize squared error between the actual and desired output. All equations can be implemented external to the PCNN thus removing any need to internally modify an existing PCNN.

1.3 Contributions

As previously stated, the PCNN has not been used for information fusion or physiologically motivated feature extraction and no adaptive PCNN currently exist. The research contributions made in these areas are briefly reviewed below.

1. *The first PCNN-based physiologically motivated feature extraction system.* This research applies primate vision processing principles such as spatial frequency filtering, state de-

pendent modulation, temporal synchronization, competitive feature selection and multiple processing paths to create the first physiologically motivated PCNN-based image feature extraction network. This is the first PCNN-based system to simulate feature extraction and attention focus observed in the biological vision system.

2. *The first PCNN-based physiologically motivated information fusion system.* This research develops the first PCNN-based information fusion network. Physiologically motivated information fusion theories are analyzed and implemented in this network. The network is used to fuse the results of several object detection techniques to improve object detection accuracy. The feature extraction and object detection properties of the image fusion network are demonstrated on mammograms and forward looking infrared (FLIR) images. The network removed 46 percent of the false detections while removing only seven percent of the true detections in the mammograms and removed 93 percent of the false detections without removing any true detections in the FLIR images. This portion of this dissertation research has been accepted for publication in *IEEE Transactions on Neural Networks.*

3. *The first adaptive PCNN.* Using gradient descent-based backward error propagation, this research develops the first fully adaptive PCNN. Given only an input and a desired output, the adaptive PCNN finds all parameter values necessary to approximate that desired output. The adaptive PCNN automatically adapts parameter values to minimize mean squared error between the actual and desired output. To demonstrate its usefulness, the adaptive PCNN was used to segment magnetic resonance images (MRI) of the brain. Adaptation was used to find parameter values that would cause the PCNN to approximate two MRI segmentation processes used in model-based vision research (2). For a given set of MRI images, the adaptive PCNN reproduced the results of the first process with 100% accuracy and approximated the more difficult second process with 90% accuracy.

1.4 Dissertation Organization

This dissertation is organized into six chapters. The following chapter provides background information on the PCNN and the primate vision system. The PCNN information presents the high level PCNN architecture, a detailed explanation of the PCNN neuron, the function of each PCNN parameter, and the physiological motivation for its unique characteristics. The vision system section develops and presents a model of the primate vision system. Based on current experimental and theoretical knowledge, this model presents the information flow and processing believed to exist in the system. Key vision processing principles described in this model are applied in Chapters III and IV to design an object detection system.

Chapter III. Feature extraction is modeled using the principles of spatial frequency filtering (3, 20, 30, 62, 72, 88), competitive feature selection (38, 60, 25, 66, 40, 39, 12, 36, 35, 37), and state dependent modulation (23, 64) which experimental data suggest exist in the primate vision system. The model is implemented using the PCNN and Gabor filters. Feature extraction is demonstrated on gray-scale images. Physiologically motivated focus of attention is added to the system and demonstrated.

Chapter IV. Theorized and experimentally confirmed cortical information linking schemes, such as state dependent modulation and temporal synchronization (32, 23, 64) are used to develop a visual information fusion network. The network is used to fuse the results of several object detection techniques. The object detection capability of the network is demonstrated on 30 mammograms and 50 FLIR images. The detection and false alarm rate of the PCNN based network is compared to rates of other published detection techniques using these real world images.

Chapter V. A mathematically equivalent model of the PCNN neuron is developed. From this model, adaptation equations are derived for all PCNN parameters. Additional PCNN knowledge is used to place the equations in a form that is suitable for application after PCNN processing is

complete. Adaptation is individually demonstrated for each parameter, and the entire adaptive PCNN is demonstrated on actual MRIs.

Chapter VI. The final chapter summarizes key conclusions and lists the individual contributions made throughout this research effort.

II. Background

The first section of this chapter presents a tutorial on the PCNN's architecture, parameters, and function. The biologically observed phenomenon of state dependent modulation and temporal synchronization are presented first as a foundation for selecting the PCNN for use in vision modeling. The PCNNs modulatory pulse-based linking is discussed in detail. The second section of this chapter develops a vision model based on experimental and theoretical data on the primate vision system. The model is simplified to contain only the information necessary to support the new information extraction and fusion approaches being developed.

The primate vision processing principles, such as state dependent modulation, temporal synchronization, and multiple processing paths described in this chapter are implemented in later chapters using the PCNN. The PCNN's modulatory pulsed-based linking capability explained in this section is used in Chapters III and IV to simulate state dependent modulation. The PCNN's segmentation capability, which is due to pulse synchronization, is used to simulate biological temporal synchronization.

2.1 The Pulse Coupled Neural Network (PCNN)

2.1.1 Overview. The PCNN is a physiologically motivated artificial neural network composed of artificial spiking neurons interconnected via multiplicative links. This artificial neural network is selected for use in this research because it contains the modulatory pulse-based linking and pulse synchronization needed to simulate the temporal synchronization and state dependent modulation observed in the primate visual cortex. The PCNN is used to implement a feature extraction and image fusion network based on these primate vision processing principles.

2.1.2 Physiological Motivation for Pulse-based Linking. As is discussed in the vision section of this chapter, the primate vision system separates the information contained within a visual image into various visual features (97). There is no known single place in the brain where these

features (orientation, color, form, texture, motion, etc.) are brought back together and combined. Many current theories propose that the neuronal pulses that transport these features synchronize in a way which associates the information to represent the original object (32, 23, 67, 92, 22, 21).

2.1.2.1 Temporal Synchronization. In 1987, stimulus-related neural oscillations were discovered in the primary visual cortex of cats (32, 23, 22, 21). These findings together with theoretical proposals (13, 9, 10, 91, 8, 82, 84, 76, 94, 24) support the hypothesis that neuronal pulse synchronization might be a mechanism that links local visual features into coherent global percepts. Two types of synchronization have been theorized, stimulus-forced synchronization and stimulus-induced synchronization. The first type is a direct result of the input stimulus. It is not oscillatory, but follows the time course of the stimulus transients. This type of synchronization is believed to play a major role in all areas of the visual cortex. The second type, stimulus-induced synchronization is believed to be produced via a self-organizing process among local neural oscillations that are mutually connected (32).

It is believed that stimulus-induced synchronization mainly supports the formation of more complex, "attentive percepts" that require iterative interactions among different processing levels and memory (23). Visual segments that are related in some fashion will synchronize and pulse in unison. These synchronized segments represent objects, or segments of objects within a visual scene. This segmentation provides objects through which dissimilar features can be associated. Gray and Singer theorize that this association is performed by temporal synchronization (32). Through this synchronization, the visual image is represented as an ensemble of synchronously pulsing objects.

This is a key concept and is used throughout this research. This concept is applied to segment and combine information in Chapters III and IV. The modulatory linking and pulse synchronization inherent to the PCNN is used to simulate both types of temporal synchronization described above.

2.1.2.2 State Dependent Modulation. Biological studies show the vision system performs substantial editing to focus attention and de-emphasizes irrelevant information (64). Even at early stages of processing, preference is given to elements to which the observer is paying attention. The response of many neurons double when the stimulus is a target of attention. State dependent signals are believed to be the stimulus that causes this preferential treatment. These are signals that originate from visual areas other than the retina, and are believed to modulate a neuron's response to any object upon which attention is focused. The signals may originate from areas in the visual cortex, or from the higher processing areas in the parietal and temporal lobes. The signals modulate a neuron's response to a stimulus within its receptive field causing a state of focused attention on the object causing the stimulus. This phenomenon is called state dependent modulation and is a method for one area of processing to superimpose its findings, or expectations on another area (64). The modulatory effect of state dependent modulations are believed to focus attention by elevating the perception of objects of interest effectively suppressing unneeded information in a visual scene.

This is a key concept and is applied throughout this research. This concept is used in Chapters III and IV to transfer information between processing units. The modulatory linking inherent to the PCNN is used to simulate state dependent modulation.

2.1.3 The Eckhorn Neuron. The PCNN uses the Eckhorn model spiking neuron (23), shown in Figure 1. The Eckhorn neuron models the pulse height, duration, repetition rate, refractory period, and modulatory inter-neural linking observed in biological dendrites. The most notable aspects of the Eckhorn neuron are the dendritic branch and the pulse generator sections. The dendritic branch contains feeding inputs which are modulated by linking inputs. Each input contains a leaky integrator which models a dendritic synapse. The leaky integrator converts incoming pulses into a persistent signal. The time constant (τ_F or τ_L) of the leaky integrator models the decay rate of neurotransmitters within the synapse. The pulse generator section is an oscillator

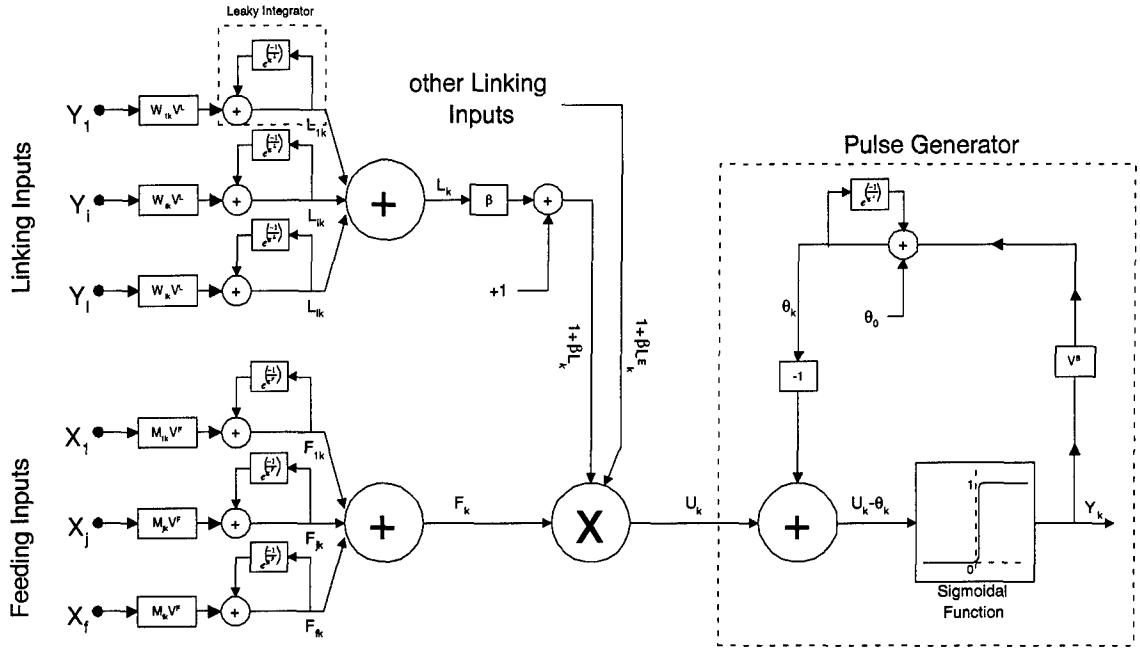


Figure 1 The Eckhorn artificial neuron used within the PCNN.

that produces an output pulse train of very short duration pulses whose frequency is based on input magnitude. The pulse generator time constant τ^S models the refractory period that occurs after a biological dendrite fires. Table 1 gives the equations required to implement a discrete time PCNN. The equations are presented in a digital filtering format, but the equation set can be rewritten in a format showing convolution (23). For detailed discussion of the PCNN see Eckhorn (23) and Johnson (46, 45, 47).

The remainder of this section describes the PCNN using simpler, but mathematically equivalent equations. To produce this description it is necessary to specify operating assumptions for the PCNN even though all terms have not been defined. All equations from this point on refer to a PCNN operating in a "pulse-once" scenario unless noted otherwise. The pulse-once scenario restricts each neuron to pulsing only once during PCNN execution. Once a neuron has pulsed, it becomes dormant and produces no additional output for the remainder of the PCNN execution. The reciprocal of the time (output pulse period) of each neuron's output pulse is used as the output frequency of the neuron. The purpose for this restriction is to remove a type of harmonic distortion

Table 1 Digital filter equations to implement a PCNN.

$$F_{jk}[n] = F_{jk}[n-1]e^{\left(\frac{-1}{\tau^F}\right)} + V^F X_j[n] M_{jk}$$

$$F_k[n] = \sum_{j=1}^{f} F_{jk}[n]$$

$$L_{ik}[n] = L_{ik}[n-1]e^{\left(\frac{-1}{\tau^L}\right)} + V^L Y_i[n] W_{ik}$$

$$L_k[n] = \sum_{i=1}^{l} L_{ik}[n]$$

$$\theta_k[n] = \theta_k[n-1]e^{\left(\frac{-1}{\tau^S}\right)} + V^S Y_k[n]$$

$$U_k[n] = F_k[n](1 + \beta L_k[n])$$

$$Y_k[n] = 1 \text{ if } U_k[n] \geq \theta_k[n] + \theta_0, \ 0 \text{ otherwise}$$

Variables

n	time index	k	counting index of neuron
i	index of neuron on linking input	j	index of neuron/pixel on feeding input
X_j	j^{th} feeding input	M_{jk}	weight applied to jk^{th} feeding input
F_{jk}	jk^{th} feeding leaky integrator output	τ^F	feeding leaky integrator time constant
Y_i	output of i^{th} neighboring neuron	W_{ik}	weight applied to ik^{th} linking input
L_{ik}	ik^{th} linking leaky integrator output	τ^L	linking leaky integrator time constant
L_k	total linking input into k^{th} neuron	F_k	total feeding input into k^{th} neuron
V^F	feeding input magnitude adjustment	V^L	linking input magnitude adjustment
U_k	total input into the k^{th} neuron	β	linking strength multiplier
θ_k	k^{th} firing threshold	τ^S	threshold leaky integrator time const.
θ_0	firing threshold offset	V^S	pulse generator magnitude adjustment
l	number of linking inputs	f	number of feeding inputs

from the PCNN output. When PCNN neurons are allowed to pulse multiple times, undesirable pulse synchronization occurs between a neuron and any neighboring neurons whose output period is a multiple of its own. This undesirable synchronization can be equated to harmonic distortion. Neurons will synchronize even if they do not meet the requirements of pulse synchronization discussed later in this chapter. To avoid this harmonic synchronization, each neuron is restricted to pulsing only once. In this scenario, pulse based synchronization satisfies the similarity definition and produces useful segmentation. Some equations for a PCNN operating in a multiple-pulse scenario will differ from the equations in this paper because of signal accumulation that takes place in the leaky integrators over time.

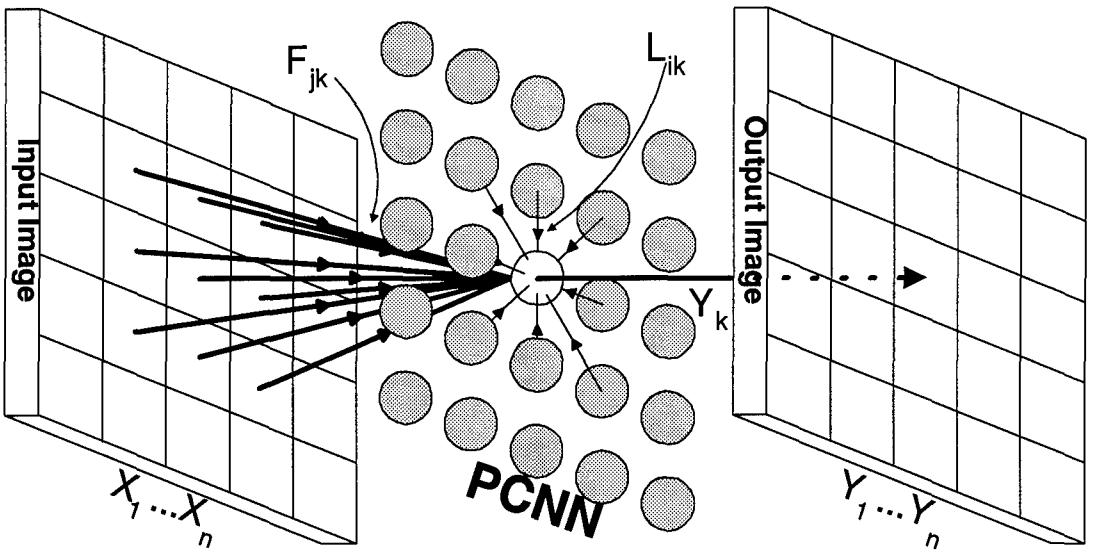


Figure 2 Example feeding and linking connections of a single neuron within the PCNN.

2.1.4 PCNN Architecture. Figure 2 shows an example of the feeding and linking connections within the PCNN. Only the external connections of a single neuron are shown for clarity. Every neuron in the PCNN would have external connections identical to those shown in the figure. This figure shows the PCNN connected to a digital image as an input and produces a digital image as an output. This is the most common connection architecture used in this research. Alternatively, the PCNN could receive its input from another PCNN, and/or send its output to another PCNN. When PCNN inputs are connected to persistent sources, such as image pixels, the leaky integrators on those inputs are omitted since their function is not needed.

When processing digital images, the PCNN typically contains one neuron for every pixel in the image. A single feeding input (F_{jk}) of the (k^{th}) neuron is connected to a spatially corresponding image pixel (X_k). Often, each neuron contains many feeding inputs which are connected in a symmetrical pattern to pixels surrounding the neuron's spatially corresponding pixel as shown in Figure 2. Typically, the feeding connections are connected to all surrounding neurons within some predetermined radius, known as the feeding radius.

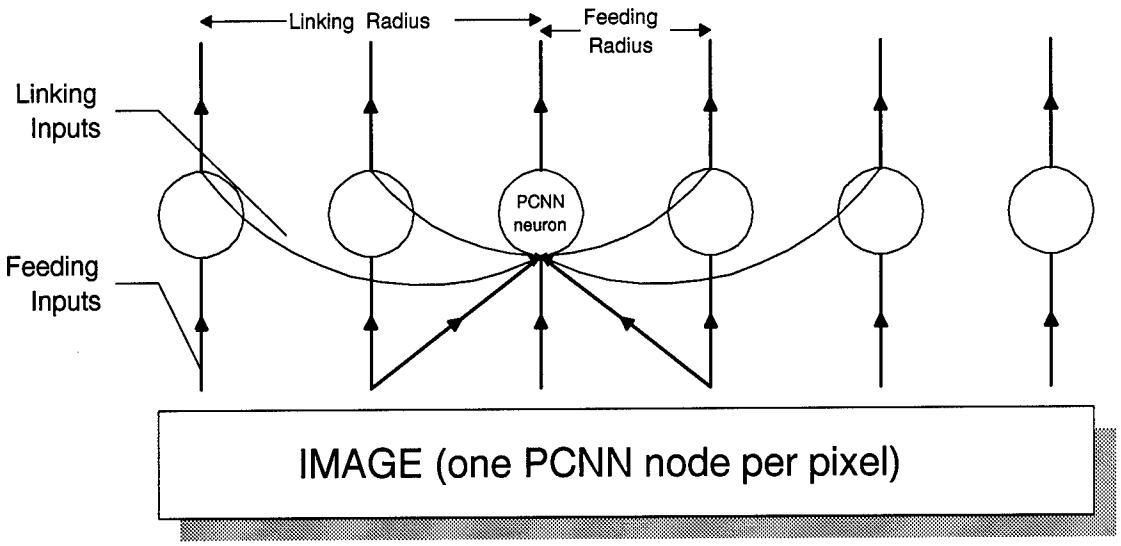


Figure 3 Sideview of example feeding and linking connections of a single neuron within the PCNN.

Figure 3 shows a side view of a PCNN. The linking inputs (L_{ik}) of each neuron are connected to the outputs (Y_i) of neighboring neurons as can be seen in the figure. These linking connections are typically connected to the outputs of all neighboring neurons within some predetermined radius. This radius is know as the linking radius. The linking connections carry the pulsed signals which are responsible for pulse synchronization (discussed in section 2.1.6).

The output of the PCNN is a pulsed signal. Each neuron output signal (Y_k) is converted to an intensity proportional to its pulse frequency. This intensity is used as the intensity of the corresponding (k^{th}) pixel in the output image.

2.1.5 The Function of the PCNN Parameters. The PCNN contains eight constants and two sets of weights. These parameters perform the following three general functions.

1. Scaling inputs from feeding and linking inputs (linking strength (β), feeding weights (M_{jk}), linking weights(W_{ik}).
2. Scaling internal signals to a desired range (magnitude adjustment constants for feeding inputs (V^F), linking inputs (V^L), and pulse generator (V^S)).

3. Adjusting the conversions between pulses and magnitudes (time constants for feeding input leaky integrators (τ^F), linking input leaky integrators (τ^L), and pulse generator (τ^S), and firing threshold offset (θ_0))

The sections below briefly describe the role of each PCNN parameter. For notational simplicity, the k subscript is omitted on all variables. All described variables belong to the k^{th} neuron.

2.1.5.1 The Pulse Generator Leaky Integrator Time Constant (τ^S). The time constant τ^S of the leaky integrator in the pulse generator section controls the resolution of the PCNN output. The value of τ^S determines the number of distinct output pulse periods the pulse generator can produce. This parameter is positive and can be equated to bandwidth. The PCNN processes input values as if they were in a sorted list. The PCNN starts by processing the input values with the largest magnitudes, then moving to input values of lower magnitudes. τ^S controls the range of values processed at each step through the list by controlling the amount the firing threshold decays during each unit of time. A decision to pulse or not to pulse is made by each neuron at each time step during PCNN execution. A larger value of τ^S causes the firing threshold to decay less during each time step. A larger time constant allows for more timesteps to occur over a given range of input values (e.g., 1.0 to 0.9). This greater number of timesteps allows the generation of more pulses to represent that input range. This provides a finer output resolution which equates to a greater output bandwidth.

As the PCNN starts operation at time $t = 0$, the value of the firing threshold θ for each neuron is at 0 which causes each neuron to pulse regardless of the magnitude of its input. Within the pulse generator, this unit area output pulse is rescaled by the magnitude adjustment constant V^S and fed back to charge a leaky integrator. The output of this leaky integrator is the firing threshold (θ). This scaled pulse charges the leaky integrator which causes the firing threshold to rise to the value V^S before time $t = 1$ is reached. For this reason, $\theta = V^S$ is considered the initial condition of each PCNN neuron, and time $t = 1$ is the first timestep an output pulse can be generated. Given

this initial condition and a pulse-once scenario, the pulse generator maps each value of its input (U) to an output pulse period of

$$T = \lceil -\tau^S \ln(\frac{U - \theta_0}{V^S}) \rceil \quad \text{for} \quad \theta_0 < U \leq V^S \tag{1}$$

$$T = 1 \quad \text{for} \quad U > V^S$$

$$T = \infty \ \text{(no pulse produced)} \quad \text{for} \quad U \leq \theta_0$$

where $\lceil \cdot \rceil$ is the ceiling operator which rounds up any fractional number to the next largest integer, U is the input magnitude into the pulse generator, θ_0 is the firing threshold offset, and V^S is the pulse generator magnitude adjustment constant. A single pulse cannot have a pulse period, but the initial condition of all neurons pulsing at $t = 0$ provides a second pulse from which a period can be deduced. Since the initial pulse is at $t = 0$, the pulsing time of the pulse produced during execution is also the pulses period.

2.1.5.2 The Global Linking Strength (β). The parameter β is a single constant that controls how the pulse period of a neuron is influenced by the output of neighboring neurons. It scales the total linking input value before that value modulates the feeding input. Larger values of β causes greater pulse synchronization.

2.1.5.3 The Linking Weights (W). The linking weights (W) scale the magnitudes of the linking pulses received from neighboring neurons. Each linking weight is independent of all other weights in the neuron. The linking radius is the distance in any direction that a neuron has linking connections to neighboring neurons. Often the linking radius corresponds to the number of neurons in any direction. A square linking pattern is often used since it is easily implemented. All examples presented in this disertation use a square linking pattern, but the equations are shape

independent. A large linking radius allows a large number of neurons to influence the pulse rate of a single neuron. This group influence gives a smoothing effect to the segmented output produced by the PCNN. Radius size can be compared to neighborhood averaging where a larger neighborhood produces a smoother output.

2.1.5.4 The Feeding and Linking Input Leaky Integrator Time Constants (τ^F and τ^L).

The purpose of the leaky integrators on the feeding and linking inputs is to convert a series of input pulses into a persistent signal. The PCNN segments input values based on magnitude and the leaky integrators convert pulsed inputs into magnitudes. The leaky integrators accumulate incoming pulses and produce a persistent signal which allows PCNN neurons with input pulse trains of similar frequencies to synchronize even if the pulse trains are not in phase. The magnitude of the leaky integrator output is a function of the input pulse train frequency and the leaky integrator time constant. Figure 4 shows the output of a leaky integrator with a pulse train input. The input

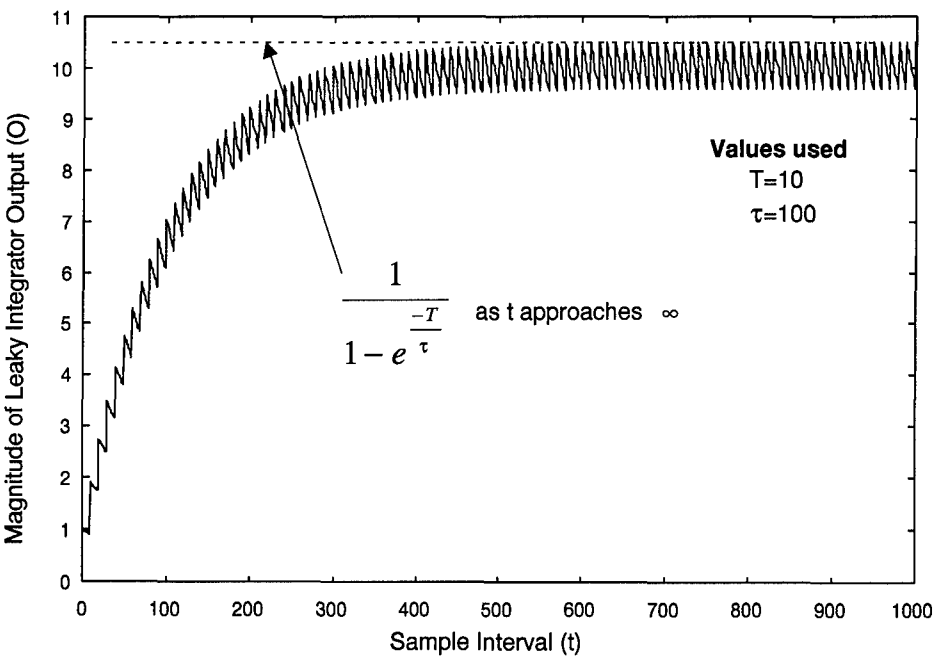


Figure 4 Output of leaky integrator with pulse train input (period=10)

pulse train has a period (T) of 10 and the leaky integrator has a time constant (τ) of 100. As time

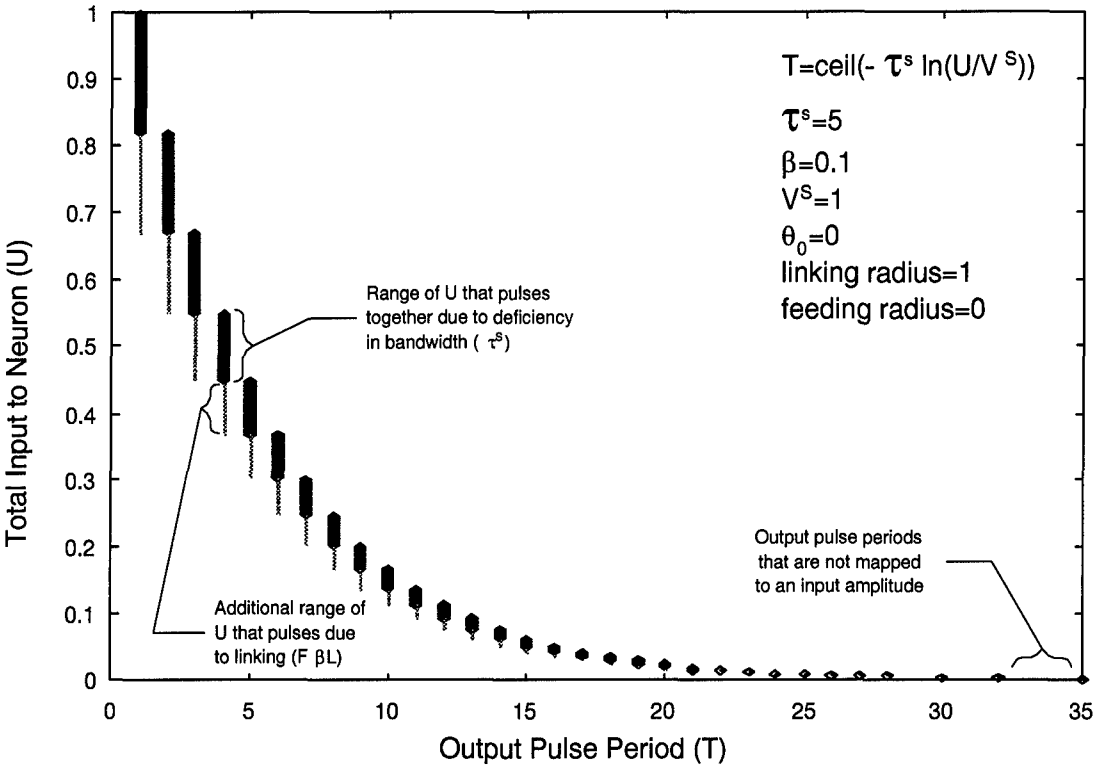


Figure 5 PCNN Segmentation (output pulse periods produced by a PCNN with a linear 0 to 1 input)

(t) becomes larger and larger, the maximum output magnitude (O) of a leaky integrator, with a pulse train input, converges to

$$O = \frac{1}{1 - \exp^{\frac{-T}{\tau}}}. \tag{2}$$

As can be seen in Figure 4, for $\tau = 100$ the leaky integrator output converges after several hundred time steps.

2.1.5.5 The Magnitude Adjustment Constants (V^F, V^L, and V^S). As shown in equation (1) and in Figure 5, the pulse generator output is a logarithmic function of its input. The pulse generator only produces a one-to-one mapping when input values are in the range from θ_0 to V^S. All inputs greater than V^S map to the pulse period $T = 1$, and all values less than or equal to θ_0 do not generate a pulse ($T = \infty$). The sum of the feeding and/or linking inputs

often exceed this range. The magnitude adjustment constants, V^F and V^L, are used to scale the magnitudes of the total feeding inputs F and the total linking inputs L, respectively, to fit within the range $[\theta_0, V^S]$. The constant V^S is used to adjust the pulse generator input operating range which effectively performs the same function as adjusting V^F and V^S.

2.1.5.6 The Feeding Weights (M). The feeding weights (M) scale the magnitudes of the feeding inputs. Each feeding weight is independent of all other weights in the neuron. The feeding radius is the distance in any direction that feeding connections exist from the center connection. As with the linking weights, square feeding connection patterns are often used to simplify implementation. Feeding weights are often adjusted to give preference to spatial characteristics of the input (spatial filtering). For example, a Mexican hat shaped weight pattern created by subtracting one 2D Gaussian from another (Difference-of-Gaussians) would give preference to objects the size and shape of the positive region of the pattern. Larger or smaller sized objects would produce a lower value on the feeding input. This concept is used in Chapter IV for the purpose of object detection.

2.1.5.7 The Pulse Generator Firing Threshold Offset (θ_0). The pulse generator firing threshold offset θ_0 provides a method of thresholding the PCNN output while it is in operation. The threshold offset is a bias value added to the pulse generator feedback loop. This bias raises the threshold by θ_0, preventing any pulse generator input value U less than θ_0 from generating an output pulse. Similar thresholding could be performed externally, but θ_0 provides a simple method for thresholding each layer when several PCNN's are connected in series.

Use of θ_0 adds unnecessary complexity to adjusting PCNN parameters. A positive value for θ_0 changes the pulse generator performance with the cost of additional processing time. As can be seen in equation (1) the pulse generator input value U is effectively shifted θ_0 in the negative direction which causes a θ_0 size portion of the pulse generator input range to remain unused. This unused range is processed needlessly unless the PCNN timeline is altered. For example, with $\tau^S = 100$ and

$\theta_0 = 0$ a pulse generator operating over the range [0,1] will pulse at timestep $t = 1$ with an input value of 1. Changing to $\theta_0 = 0.6$ causes that same input to generate a pulse at timestep $t = 92$ with no pulses occurring during the first 91 timesteps. The magnitude adjustment constants can be adjusted to compensate for the unused timesteps, but become interdependent with θ_0 which complicates their adjustment. In most cases, the thresholding performed by θ_0 can be performed with less complexity outside of the PCNN. The parameter θ_0 is not used in this research. All references to it are to provide equations that accurately describe the neuron model.

2.1.6 Pulse Coupling Performs Temporal Synchronization. Pulse-based synchronization is the key characteristic that distinguishes the PCNN from other types of neural networks. Pulse synchronization provides a segmentation property useful in image processing. Neighboring neurons with similar inputs pulse in synchrony to represent a segment of the input image. Neurons with similar feeding input characteristics (color, intensity, etc.) have similar pulsing rates. The linking connections cause neurons, in close proximity and with related characteristics, to pulse in unison (synchronization) (32, 23). The PCNN synchronizes neurons base on similarity. This similarity is defined by the magnitude of the total input (U) of a neuron relative to the magnitude of the total input of neighboring neurons within its linking radius. When using a digital image as an input, these input magnitudes are the values of the image pixels. The pixel values could be a measure of brightness, a filter's response, a color value, or any other measurement represented at each point in the image. A neuron is similar to any neuron within its linking radius that has an input magnitude within $F\beta L$ greater than its own, where F is the total feeding input value to the neuron, L is the total linking input value, and β is the value of the linking strength between neurons. For explanation purposes, assume each PCNN neuron has a single input. This forms a one-to-one relationship between neurons and pixels (i.e., each neuron represents one pixel). A pixel is similar to any neighboring pixel that has a magnitude within $F\beta L$ greater than its own. Shown

in equation form, a pixel with a magnitude of $I1$ is similar to a pixel with a magnitude of $I2$ if

$$0 \leq I2 - I1 \leq F_{I1}\beta L_{I1} \tag{3}$$

Because of the multiplicative linking connections, this relation is not as simple and straight forward as it first appears. The value of $I1$ and $I2$ are dependent upon the pulsing activity of neighboring neurons which makes them dependent upon one another. The following discussion makes some simplifying assumptions to demonstrate the complexity of determining which neurons are similar.

The pulse period (T) of a digitally simulated neuron with constant linking inputs is defined by the equation

$$T = \left\lceil -\tau^S \ln\left(\frac{U}{V^S}\right) \right\rceil \tag{4}$$

. As previously stated, U is the total input to the neuron which is defined as $U = F(1 + \beta L)$, τ^S is the pulse generator leaky integrator time constant, and V^S is the pulse generator magnitude adjustment constant. Without any linking inputs ($L = 0$), bandwidth limitations of the neuron (controlled by the value of τ^S) would cause input values between 0 and 1 to fire in non-overlapping logarithmic sized groups as shown in Figure 5 (much higher values of τ^S are typically used). Notice that if $L = 0$, U is equal to the total feeding inputs F. The scale of the output pulse period axis is time units where one unit is the maximum pulse firing rate the neuron bandwidth will support. For a digital implementation, each unit would be one time-step on the simulation clock. The values of U that pulse each time slice without linking present are shown by the bold lines. The set $P(t)$ is defined to be the values of U that pulse at time t when no linking is present. Adding a constant linking input to a neuron extends the lower limit of $P(t)$ by $F\beta L$ (shown as the thin line in Figure 5). We define the set S of real numbers that are added to $P(t)$ due to linking to be the synchronization range of a PCNN neuron,

$$S = [F, F + F\beta L]. \tag{5}$$

This synchronization range defines the similarity in pixel intensity which will cause the output pulses of neurons to synchronize. A neuron that would not normally fire at time t will fire in synchrony with other neurons that fire at time t if

$$S \cap P(t) \neq \emptyset. \tag{6}$$

This criteria must be met for a neuron to synchronize with other neurons pulsing at a particular pulse frequency.

Notice in Figure 5, the total pulse range $(P(t) \cup S)$ for each time t overlaps the total pulse range for time $t+1$. This means a neuron with a value U in the overlapping region can fire at either time t or $t+1$ depending on linking inputs. So will a particular neuron N_i fire at time t or $t+1$? Expanding the earlier assumption of a constant linking input signal to state the linking inputs originate as the constant outputs of neighboring neurons as shown in Figure 2, makes L a function of the feeding and linking inputs of neighboring neurons. Since the value of L originates as the output of neighboring neurons and the synchronization range S is a function of L, Equation (5) implies segmentation is image content dependent. For two adjacent neurons that are linked, the output of each neuron is dependent upon the output of the other. Since linked neurons are dependent upon one another, finding the output pulse period of a particular neuron requires solving simultaneous equations. For example, the output period of nine neurons connected in a 3×3 PCNN is described by the following matrix equation,

$$\begin{bmatrix} T_1 & T_2 & T_3 \\ T_4 & T_5 & T_6 \\ T_7 & T_8 & T_9 \end{bmatrix} = \begin{bmatrix} \lceil \tau^S \ln(F_1(1+\beta L_1)) \rceil & \lceil \tau^S \ln(F_2(1+\beta L_2)) \rceil & \lceil \tau^S \ln(F_3(1+\beta L_3)) \rceil \\ \lceil \tau^S \ln(F_4(1+\beta L_4)) \rceil & \lceil \tau^S \ln(F_5(1+\beta L_5)) \rceil & \lceil \tau^S \ln(F_6(1+\beta L_6)) \rceil \\ \lceil \tau^S \ln(F_7(1+\beta L_7)) \rceil & \lceil \tau^S \ln(F_8(1+\beta L_8)) \rceil & \lceil \tau^S \ln(F_9(1+\beta L_9)) \rceil \end{bmatrix} \tag{7}$$

where F_i is the total feeding input into the i^{th} neuron and L_i is the total linking input into the i^{th} neuron. Since the value of each L_i is composed of the outputs of neighboring neurons (see Figure 2),

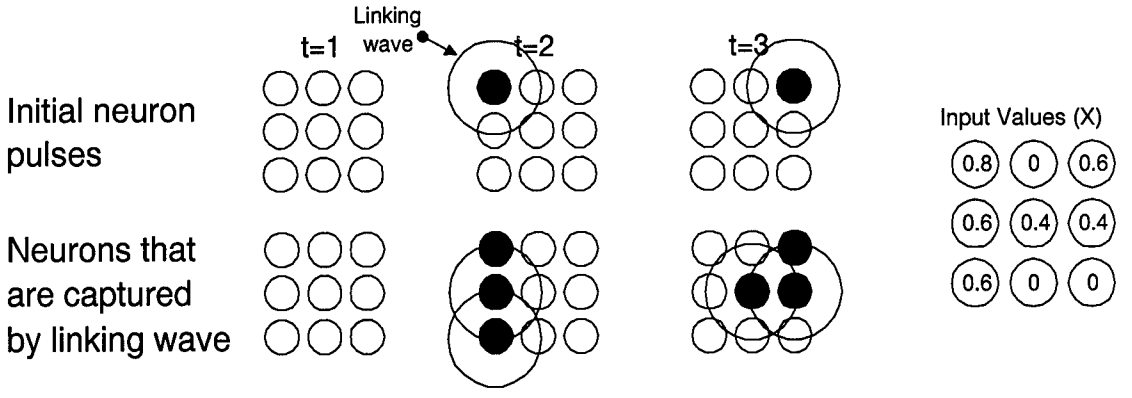


Figure 6 Firing sequence of PCNN neurons due to linking.

the value of each T_i is dependent upon the values of other T_i's. Finding the output period of any single neuron requires solving the nine equalities simultaneously. In essence, this is what the PCNN does. The assumption of a constant linking input simplifies the problem significantly. Since the PCNN is based on a spiking neuron, all linking signals are pulses which means linking inputs are not constant. The actual operation of the PCNN is more complex than this simplified example, but the functional concept is the same.

The actual PCNN solves the inter-neuron dependencies in a unique way. No linking signals are present until the first neuron fires. The pixels with the largest magnitude within an input image cause their corresponding neurons to fire first. This firing initiates a linking signal (linking wave) which travels through the multiplicative linking interconnects causing other neurons with similar inputs to fire (46).

Figure 6 shows the pulsing sequence of a sample 3×3 neuron PCNN. Dark circles represent neurons that pulse during that timestep, light circles represent neurons that do not pulse. Only the first three timesteps of PCNN execution are shown. Using arbitrary PCNN parameter values, a linking radius of 1, and the input values shown in the figure, the upper left neuron pulses first (at time t=2) since it has the greatest input magnitude. This output pulse travels through the linking connections to neighboring neurons. This linking signal flow is called a linking wave. This linking

wave increases the L_i value of all neighboring neurons. If the total input magnitude $F_i(1+\beta L_i)$ of any neighboring neuron exceeds its firing threshold, those neurons will pulse producing a linking wave of their own. Any neuron that would not normally pulse at a particular time, but pulses due to a linking wave is said to be captured by the neuron that emitted the wave. All neurons that pulse together due to linking are considered a single group. This grouping effectively segments an image into objects. Note the upper right neuron did not pulse with the first group because it was not within the linking radius of any pulsing neuron. The neuron has the same input as neurons that did pulse, but the neighbor requirement of the similarity definition was not met; thus it was not similar to the first group.

Since linking fields overlap, grouping occurs beyond the limits of a single neuron's linking radius. A single neuron can fire and cause a domino effect that continues until all neurons with similar inputs fire in phase synchrony with the first neuron. This group of synchronously firing neurons represents a distinct segment within the image. The segmentation process repeats each time step, on neurons that have not fired, until all neurons within the PCNN have fired and the image is completely segmented.

2.2 A Model of the Primate Vision System

2.2.1 Overview. This section develops a vision model based on experimental and theoretical data concerning the primate vision system. The purpose of this section is to provide necessary background of the information processing and fusion techniques used within the biological vision system. A high level vision model is developed by first stating various facts and hypothesis about the visual system and then developing a model that incorporates these facts and hypothesis. The model is described using high level diagrams which are expanded and decomposed to the point that a physiologically-based neural network can be used to implement the key information processing

concepts. The model is simplified to contain only the information necessary to support a new information extraction and fusion approach.

To assist the reader, several vision topics have been placed near the research where they are used. The primate vision principles of state dependent modulation and temporal synchronization were discussed earlier in this chapter. The discussion of the detailed processing which occurs within individual functional areas is presented in Chapter III where it is used for feature extraction. The key topics in this chapter are the concepts of multiple information paths and the concept of areas sending information to unrelated areas to assist in processing.

2.2.2 Pathways and Functional Areas. Despite the enormous complexity of the primate cortical visual system, studies suggest it can be modeled by two basic hierarchical pathways, the parvocellular pathway and the magnocellular pathway (96). The former pathway predominantly processes color information, and the later processes form and motion.

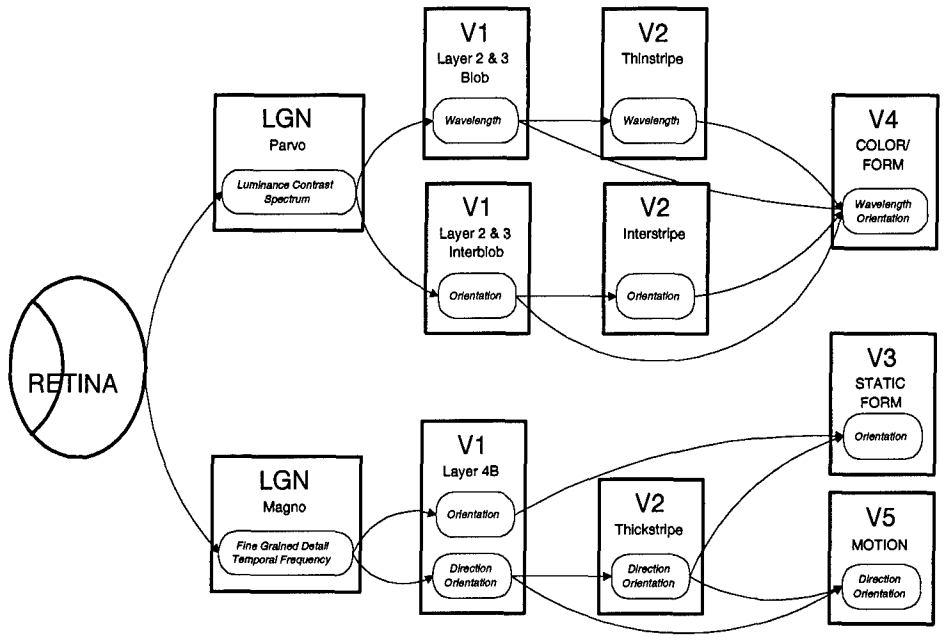


Figure 7 Forward information flow of the visual system model

Figure 7 shows a model of these two pathways. The entry point of an image into the model is the retina. The biological retina has luminance and color detectors which interpret light images and preprocesses the image before relaying it to the rest of the visual system. The area marked LGN models the biological lateral geniculate nucleus. This area separates the retinal image into fundamental components such as luminance, contrast, frequency, etc.. The areas of the model labeled with names starting with the letter V model specific areas in the human visual cortex. Each of these areas is believed to maintain one or more processed, but topographically correct images of the light pattern that falls upon the retina (97). The processing that is applied to the image is discussed later in this section. Area V1 represents the striate visual cortex and is believed to contain the most detailed and least processed visual image found in the cortical visual areas (V1,...,V5). Henceforth, the visual image maintained by each visual cortex area is referred to as a visual map, or simply a map. Area V2 contains a visual map that is less detailed and more processed than area V1. Areas V3, V4, and V5 are called specialty areas because it is believed that they process only selective information such as form, color, and motion, respectively. The maps maintained within the specialty areas are less detailed than the map within V2 and only reflect the particular information each area processes. For example, the visual map in area V3 would predominantly contain information about the form contained in the image that is present on the retina (97, 90). It would contain little or no color or motion information.

The names within the LGN, V1, and V2 boxes in Figure 7 refer to functionally distinct sections of the area. Parvo, magno, blob, interblob, thinstripe, thickstripe, and interstripe are all terms used by early researchers to describe subsections of the visual areas that are visually distinct in appearance. These terms are still in use today and are included to link the vision model to the biological vision system.

2.2.3 Processing Hierarchy. Information flows in both the forward and reverse directions in a hierarchical fashion within the vision system. A portion of the forward flow of the orientation

processing pathway is implemented in the following chapter. The reverse information flow is not directly implemented in this research, but a mechanism for future implementation is developed. The principle of state dependent modulation is observed throughout the vision system and is believed to be used to perform some of the information processing and transfer discussed in this chapter. The key concept in this section is that information from a processing area can be used to assist in the processing of another area that processes a completely different type of information. This concept is used in the information fusion chapter and feature extraction chapters.

2.2.3.1 Forward Visual Information Flow. Each box in Figure 7 represents a distinct visual map believed to be maintained in the respective portion of the visual area (97). The ovals denote the specific type of information contained within each map. The visual areas are almost fully connected which is not shown in the diagram. For clarity, the diagram shows only the stronger connections which are pertinent to the model being developed. The results of the processing performed by each area is sent to the next area in the hierarchy to be incorporated into its map.

As you move to the right in the processing hierarchy shown in Figure 7, the spatial area processed by each processing unit increases (97). For example, a single neuron in V3 processes a larger part of the input image than a single neuron in V1. The orientation processing path of the dynamic form pathway will be used to demonstrate the increasing size of receptive fields (Figure 8). This figure is constructed from existing experimental and theoretical data (97, 90, 38, 12). The top row of Figure 8 shows the forward flow of this orientation processing pathway. Each visual area is shown processing the letter A. The size of the receptive fields of the processing unit in each area is shown by the ellipses in the second row (note, the receptive fields are not drawn to scale, but are merely used to demonstrate a concept). The third row shows a possible output of the processing units which is communicated to other areas (97, 90, 38, 12). Each successive layer in the hierarchy has a larger receptive field, and produces an output based on a larger amount of information. This concept of receptive field size is used in Chapter III to explain the effects of spatial uncertainty.

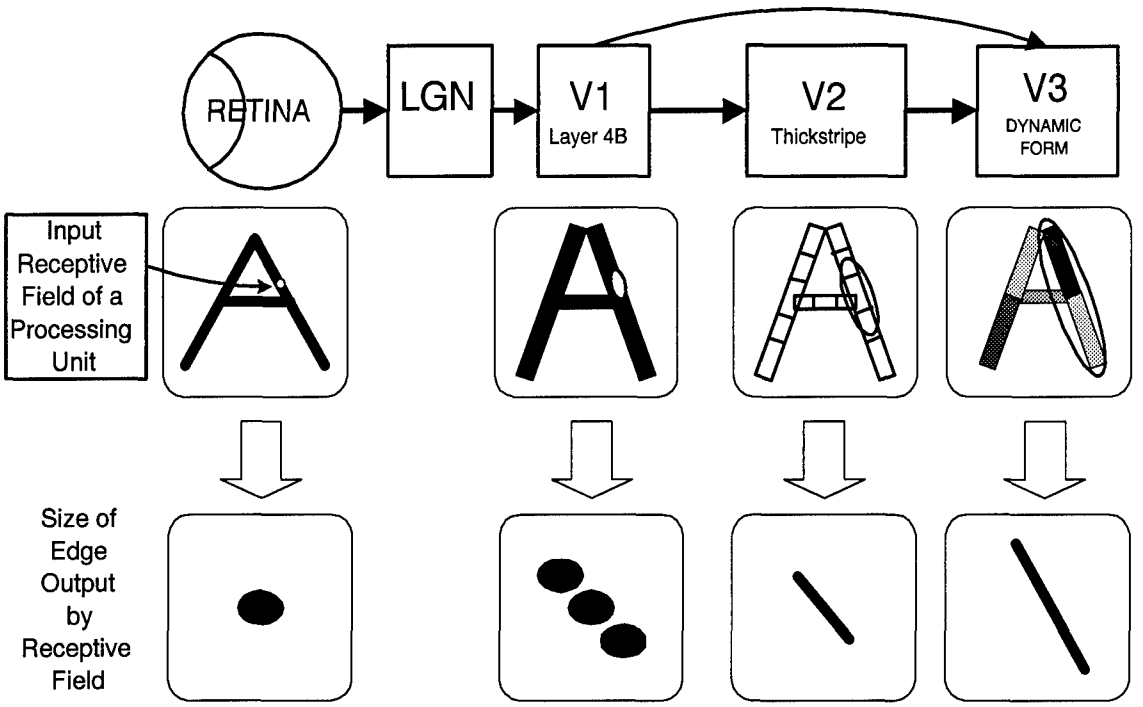


Figure 8 Example of orientation processing in the visual system model

2.2.3.2 Reverse Visual Information Flow. Zeki theorizes that "the most precise map is area V1, followed by V2. The specialized areas (V3, V4, V5) must therefore send information back to V1 and V2 so that the results of the processing can be mapped back onto the visual field" (98). These feedback connections are called reentrant connections. Figure 9 shows the reentrant connections used to transfer information back into the maps of related areas. Each connection does not necessarily carry the same type of information. This is due to the fact that the receptive fields of the processing units within a hierarchical level are larger than the receptive fields of the units found in a previous hierarchical level. The forward projections of information are patchy and discrete and the return projections are diffuse and fairly non-specific. Another function of these reentrant connections is to supply information to resolve any conflicts that may exist in a lower level (98). As Figure 9 shows, the reentrant connections from visual areas are not restricted to the areas that supply its input. It is theorized that these additional connections are used for resolving conflicts between areas that have different capabilities but are responding to the same stimulus

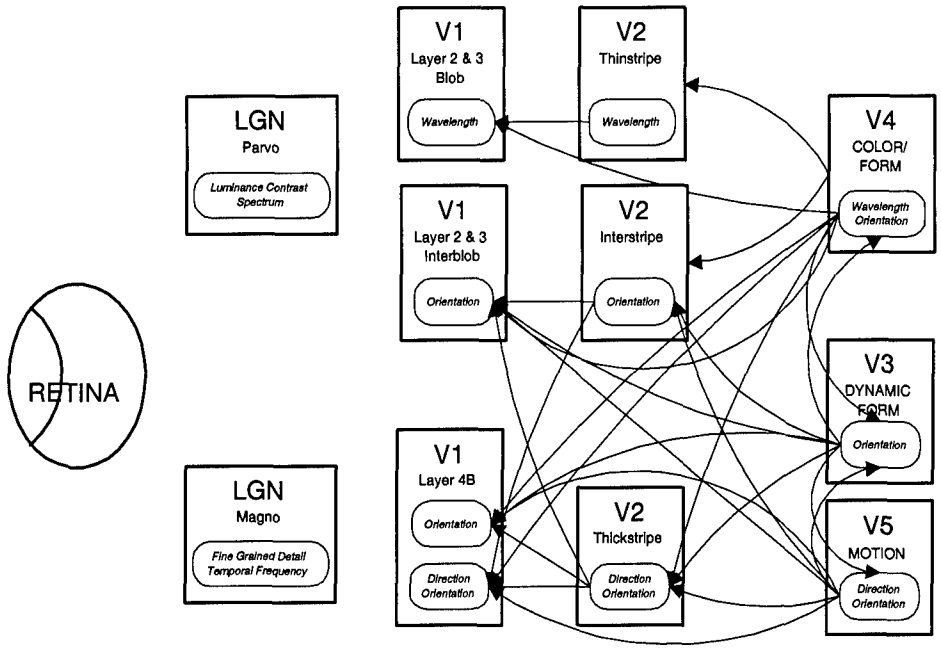


Figure 9 Reverse information flow in the visual system model

(97). For completeness, the three areas within V1 are each connected and the three areas within V2 are each connected. Also, there is a direct connection between the LGN and the three specialty areas V3, V4, and V5. These connections are omitted because their functions are either unknown or of no significance to the model being designed.

Based on theoretical and observed data, Figure 10 shows the feedback (reentrancy) of the output of each visual area into the maps of the areas in previous hierarchical levels which is believed to occur in the primate vision system (97, 90, 38, 12). The solid black ellipses shown in each map represent the size of the receptive fields of the processing units that operate on that particular map. As stated previously, the receptive field grows larger at each successive hierarchical level, and each level reenters its output information into lower levels to resolve any conflicts that may exist.

2.2.4 Information Flow. Figure 11 shows the forward and reverse information flow in the visual system which was previously presented. The ovals denote the type of information processed by each area, bold lines denote information flow in the forward direction and normal lines denote

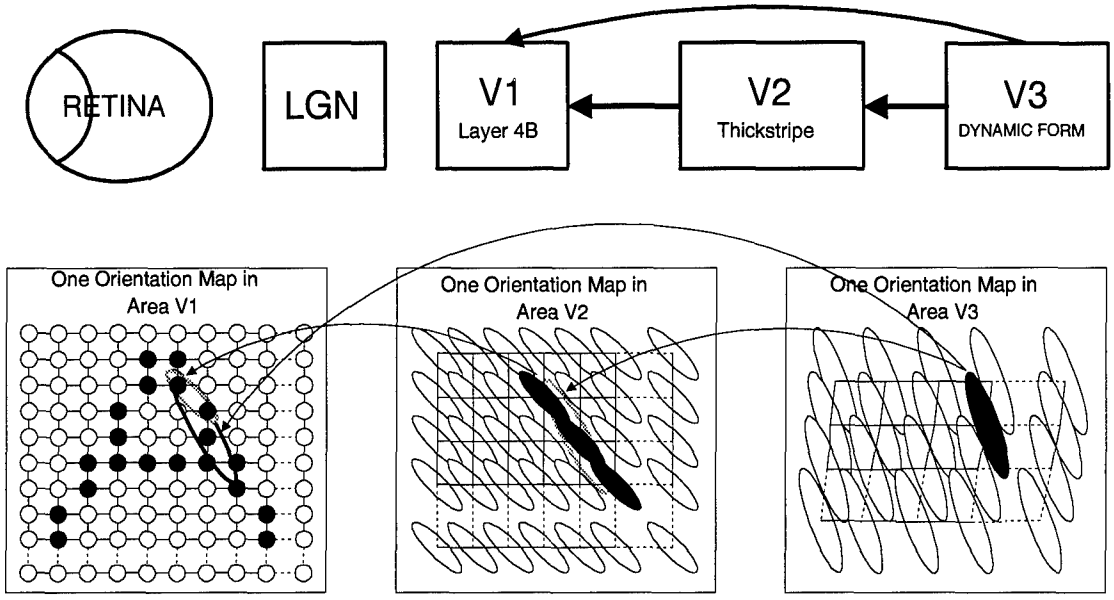


Figure 10 Feedback in orientation processing in the visual system model

reverse flow of information. The subscripts on the V1 and V2 processing areas denote the type of selective processing units located within that area (to remove ambiguity). It is important to note that many areas receive a reverse flow of information which is not the type they normally process. For instance, layer 4B of area V1 (Figure 9) contains units of cells which are primarily orientation selective. These processing units are neither wavelength nor direction selective, but still receive this type of information from areas V4 and V5. This information is not ignored, but is combined (linked) with the orientation information to remove any ambiguities or conflicts.

The dominant type of information produced by each processing area is listed in Table 2. Much is still unknown about the vision system, but this list of outputs is sufficiently complete for the purpose of this research which is to model feature extraction and information fusion.

Based on the knowledge that each processing unit is a group of neurons operating on input signals carried by axons (97), Figure 12 gives a probable model for the connections that provide the input and reentrance of information.

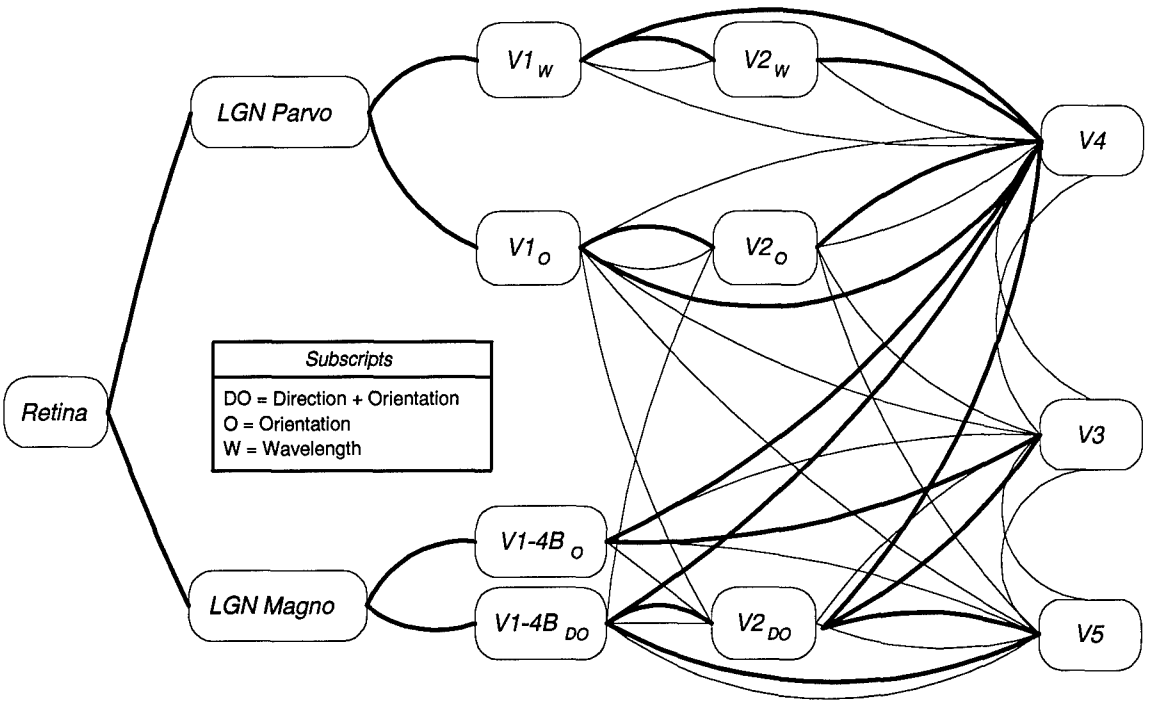


Figure 11 Information flow in the visual system model

Visual area V1 is used as an example to demonstrate how information could be reentered, but the same model would apply to all of the visual areas. Figure 12 also shows a blackbox representation of the filter used to model the neuronal process within the processing unit. Notice the filter operates on the combination of all inputs. The method used to combine the input, lateral inhibition, and reentrant signals is key to the information fusion process within the visual model. As previously discussed, state dependent modulation is used to combine information at the neuronal level. This concept is the basis for transporting and combining information throughout the vision model. State dependent modulation is implemented using the modulatory pulse-based linking found in the PCNN. The filter shown in Figure 12 is discussed and implemented in Chapter III.

Table 2 Signal definitions for the vision model

Signal within Vision Model	*Signal Type*
Output of Retina	Spectrum, Luminance, Temporal frequency
Output of LGN Parvocellular Layers	Spectrum, Luminance,
Output of LGN Magnocellular Layers	Luminance, Temporal frequency
Output of Area V1 (layers 2 & 3) which processes Wavelength	Wavelength vector
Output of Area V1 (layers 2 & 3) which processes Orientation	Orientation vector
Output of Area V1 (layer 4) which processes Orientation	Orientation vector
Output of Area V1 (layer 4) which processes Direction+Orientation	Direction + Orientation vector
Output of Area V2 which processes Wavelength	Wavelength vector
Output of Area V2 which processes Orientation	Orientation vector
Output of Area V2 which processes Direction+Orientation	Direction + Orientation vector
Output of Area V3	Set of orientation vectors
Output of Area V4	Set of color vectors
Output of Area V5	Set of motion vectors

2.3 Summary

This chapter has presented a tutorial on the PCNN and on primate vision processing. The biologically observed vision principles of state dependent modulation, temporal synchronization, and multiple processing paths are key topics used in later chapters. Theoretical and experimental data has been presented to describe their function in primate vision processing and the PCNN section discussed the modulatory pulse-based linking and temporal synchronization capabilities that are used to simulate them.

Throughout the vision section, the multiple processing paths are described and explained. The early vision processing believed to be performed in one of these paths is simulated in the following chapter using the PCNN and Gabor filters. This single processing path simulation is used to demonstrate each key vision processing principle. Modifications are described that incorporate other processing paths into the simulation.

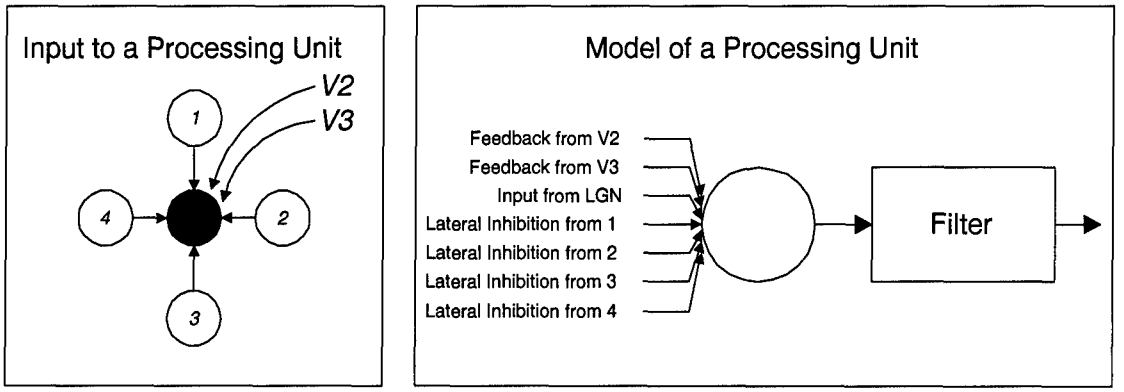


Figure 12 A processing unit within the visual system model

III. Simulated Visual Feature Extraction Using the PCNN

3.1 Overview

In this chapter, PCNNs and Gabor filters are used to simulate the biological feature extraction performed in the primary visual cortex. Substantial experimental evidence suggests that some form of spatial frequency analysis is being performed in the primary visual cortex (3, 20, 30, 62, 72, 88, 61). Studies have found orientation-selective, direction-selective and wavelength-selective cells which perform this analysis. The Gabor function has been shown to be a good model for many of these cells (15, 63, 48). A feature extraction model is designed using filters created from Gabor functions to simulate the orientation-selective cells in the biological vision system. The information produced by these filters is processed with several PCNNs to determine the pitch (magnitude along radial axis in two dimensional frequency domain), orientation, and intensity that exist at each location in the input field of view. This feature extraction model performs a spatial frequency analysis to produce simulated visual features. Though spatial frequency filters are used to demonstrate the capabilities of the the model, the model is not limited to extracting spatial frequency features. The model can be easily extended through alternate filter choices to extract color and motion features from color imagery and motion video.

All spatial frequency filters have an inherent space/frequency tradeoff that causes a degree of spatial uncertainty in the location of objects in their output. Many vision models simulate biological visual processing using spatial frequency filters, then apply digital image processing techniques to extract visual features (38, 60, 25, 66, 40, 39, 12, 36, 35, 37, 33, 75). The pixel-based digital image processing techniques used to extract features often magnify the spatial uncertainty by causing artifacts in the simulated visual features. In this chapter, the PCNN is shown to be a good alternative to these pixel-based techniques. Using the physiologically motivated principle of temporal synchronization (32, 23, 22, 21, 67, 92), the PCNN is used to form objects from the filter outputs, and determine the features that exist at each spatial location. This object-based

approach does not produce the feature artifacts that plague pixel based approaches. Example features produced by several pixel-based image processing techniques are presented and compared with features produced by the PCNN feature extraction network.

The end use of simulated visual features is typically in an object detection/recognition system. An object detection system requires a method of focusing attention on desired objects while ignoring the rest of the visual scene. Using the PCNN to implement the physiologically motivated principle of state dependent modulation, a focus of attention capability is added to the PCNN feature extraction network creating a simple object detection system. In a simple example, this focus of attention capability is used to detect a desired object within a visual scene containing several objects. In the next chapter, this simple object detection system is enhanced to include additional object detection and information fusion capabilities. The object detection capability of this enhanced system is demonstrated on x-ray and FLIR images with promising results.

The contributions in this chapter are:

1. The first use of a PCNN to perform object-based physiologically motivated feature competition.
2. The first physiologically motivated PCNN-based visual feature extraction network.
3. The first use of a PCNN to implement state dependent modulation for focus of attention.

3.2 Simulating Visual Features Using Filters

To simulate biological feature extraction, we need to simulate the processes within the biological visual processing areas (V1, V2, V3, V4, and V5) and then select features from that information. The primate vision system is a multi-stage hierarchical system of neurons which extracts features from the visual scene for the purpose of object detection/recognition (97). The early stages of visual processing (lateral geniculate nucleus, primary visual cortex, and pre-striate cortex) separate images that fall upon the retina into color, shape and motion (98). Studies of

these areas have found orientation-selective, direction-selective and wavelength-selective cells. To simulate biological feature extraction, we use the hypothesis that neuronal processing units are best described as filters that are selective along multiple stimulus directions(90). Since all images used in this research (mammogram and FLIR) are static and gray-scaled, only the biological static form pathway is discussed (Figure 7). Table 3 gives a list of possible filters that can be used to approximate each visual area of the biological static form pathway.

Table 3 Filters that can approximate functions performed in the vision model

Vision Model Area	*Possible Filter Models*
Retina (R)	Difference of Gaussians filter (12, 7), Wavelet filter.
LGN Parvocellular	Difference of Gaussians filter (12, 7)
LGN Magnocellular	Difference of Gaussians filter (12, 7)
V1 wavelength selective	2D Gabor filters (89), Gaussilinear or Wavelet filters (90)
V1 orientation selective	2D Gabor filters (89) , Gaussilinear or Wavelet filters (90)
V1 layer 4B orientation selective	Gaussilinear or Wavelet filters (90), orientation selective filters (12, 7)
V1 layer 4B orientation + direction selective	Gaussilinear or Wavelet filters (90), orientation selective filters (12, 7)
V2 wavelength selective (V2W)	2D Gabor filters (89), Gaussilinear or Wavelet filters (90)
V2 orientation selective (V2O)	Gated dipole filter (12, 7)
V3 Dynamic Form	Gated dipole filter (12, 7)

To limit the scope of this research, only the visual processing performed in the primary visual cortex (area V1) is simulated. More experimental and theoretical data exist for this cortical visual area than the others (V2, V3, V4, V5, etc.). For this reason, the primary visual cortex is the focus of the remainder of this chapter. However, the applicability of this research is not limited to this visual area and can be extended to the other visual areas as additional knowledge is amassed

Recent physiological evidence suggests that the primary visual cortex performs a spatial frequency analysis, distributing information in the scene among multiple channels which are selective to different spatial frequencies (15, 69, 97). Any of the filters in Table 3 could be used to simulate this space/frequency analysis. Since 2D Gabor filters have been found to be a good model for the 2D receptive fields of cells in the primary visual cortex (15, 48, 89), they are used for this research. These filters are used to simulate the orientation-selective cells in the primary visual cortex. The

results presented in this chapter are not filter dependent. Any spatial frequency filter could be used with similar results.

3.2.1 The Gabor Function. In 1980, a model for the receptive field of simple cells in the visual cortex was proposed which consisted of harmonic oscillation within Gaussian envelopes (15, 63). In 1984, direct measurements of cortical cells showed this model approximates cell receptive fields (48). These Gaussian damped oscillations belong to a class of functions known as Gabor functions. Gabor functions are discussed in detail in (15, 27, 48, 89).

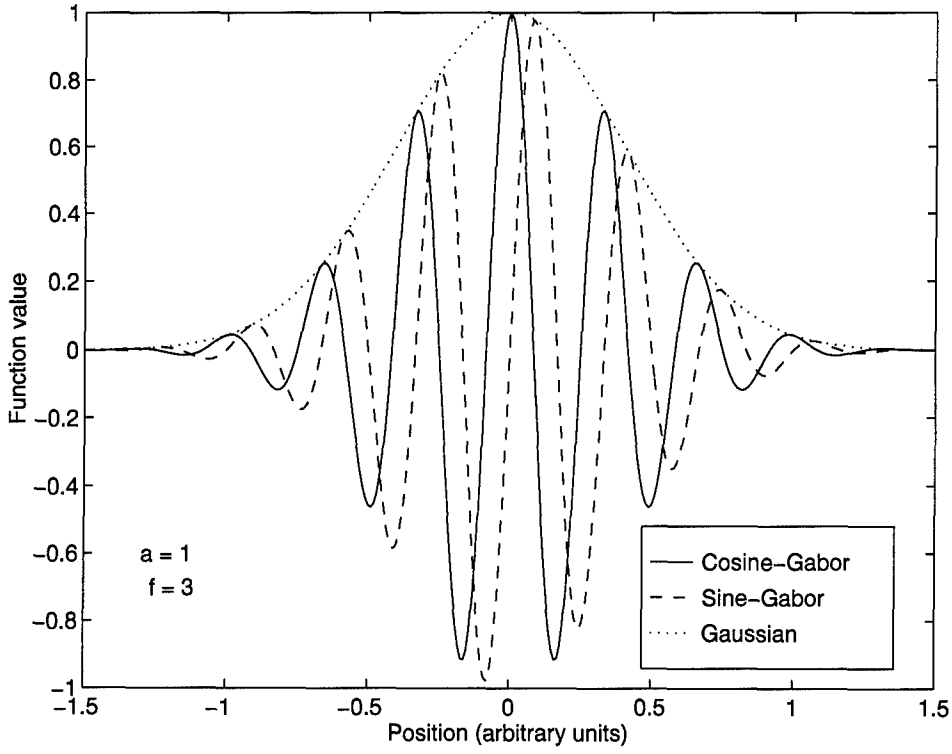


Figure 13 A one dimensional Gabor function.

Figure 13 shows a one dimensional Gabor function constructed from a sinusoidal wave within a Gaussian envelope. Both a sine wave and a cosine wave are shown as examples of the sinusoidal wave. In two dimensions, the Gaussian envelope surrounds a sinusoidal plane wave. For this

research we use the following frequency domain definition of a Gabor function (27). The cosine-Gabor function is defined as

$$G_{\cos}(f_x, f_y) = \frac{1}{2}\exp\left(-\pi\left[\left(\frac{f_y \sin\theta + f_x \cos\theta - \rho}{a}\right)^2 + \left(\frac{f_y \cos\theta - f_x \sin\theta}{b}\right)^2\right]\right) + \frac{1}{2}\exp\left(-\pi\left[\left(\frac{f_y \sin\theta + f_x \cos\theta + \rho}{a}\right)^2 + \left(\frac{f_y \cos\theta - f_x \sin\theta}{b}\right)^2\right]\right), \quad (8)$$

and the sine-Gabor function is defined as

$$G_{\sin}(f_x, f_y) = \frac{1}{2j}\exp\left(-\pi\left[\left(\frac{f_y \sin\theta + f_x \cos\theta - \rho}{a}\right)^2 + \left(\frac{f_y \cos\theta - f_x \sin\theta}{b}\right)^2\right]\right) - \frac{1}{2j}\exp\left(-\pi\left[\left(\frac{f_y \sin\theta + f_x \cos\theta + \rho}{a}\right)^2 + \left(\frac{f_y \cos\theta - f_x \sin\theta}{b}\right)^2\right]\right). \quad (9)$$

where ρ is the center radial spatial frequency, θ is the center angular spatial frequency, b is the spatial frequency bandwidth along the radial axis, a is the spatial frequency bandwidth perpendicular to the radial axis, and j is the imaginary value $\sqrt{-1}$.

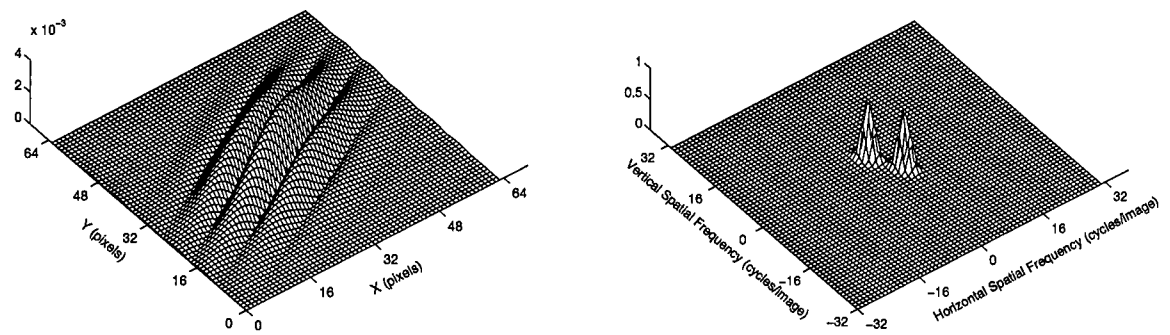


Figure 14 Two dimensional Gabor function. (a) Spatial domain response (b) Fourier transform of Gabor function.

Figure 14 shows a plot of a Gabor function in the spatial and frequency domain. The values used in this example ($\rho = 8$, $\theta = -\frac{\pi}{3}$, $a = \sqrt{\pi}\frac{\rho}{2}$, and $b = \frac{a}{2}$) produce a Gabor function with a radial center frequency of 4 cycles per image, oriented to 60 degrees, with a 1.5 octave bandwidth.

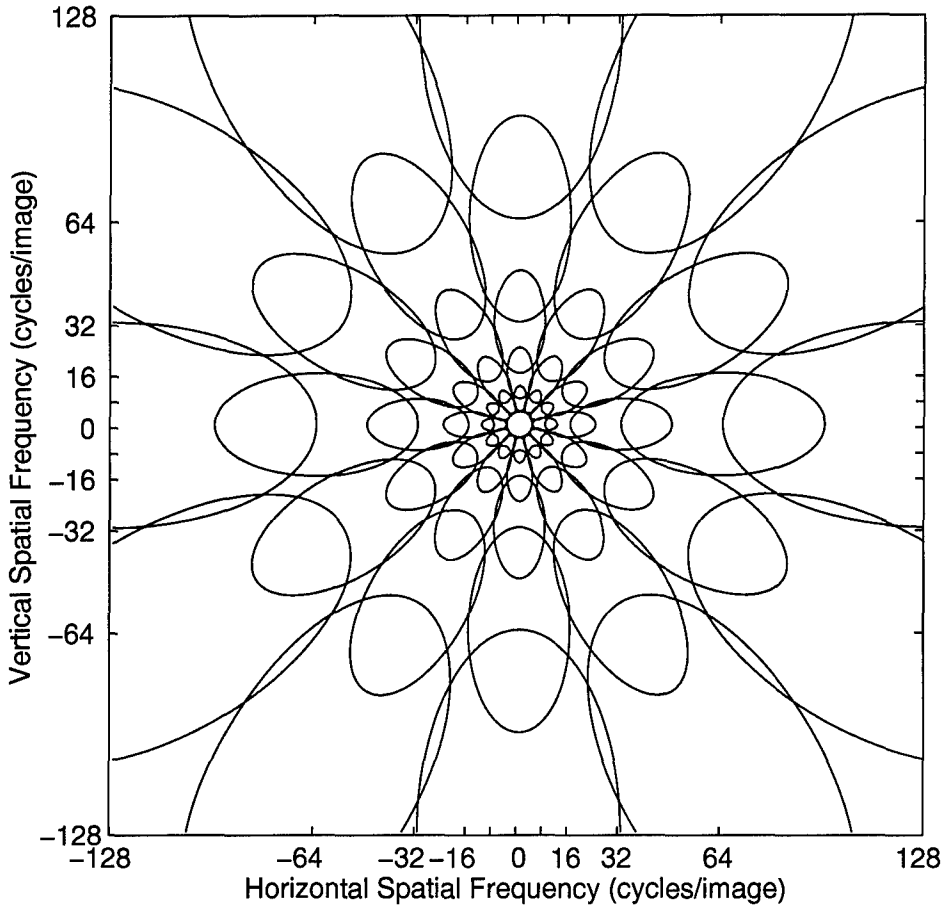


Figure 15 Frequency domain plot of spatial frequencies covered by multiple Gabor filters.

3.2.2 Extracting Spatial Frequency with Gabor Filters. Zeki theorizes the cells in the primary visual cortex are organized to form multiple views of the retinal image, each view being devoted to a different visual attribute (97). Many of these cells are selective to particular spatial frequencies. The Gabor function can be used to model these cells. The Gabor function used as a filter kernal is a Gabor filter. Many of the multiple views believed to exist in the visual cortex can be modeled using multiple images, each filtered by a Gabor filter tuned to a unique spatial

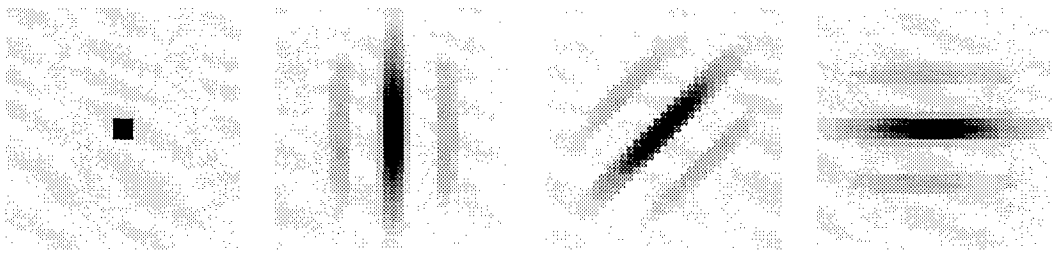
frequency. As shown in Figure 14b, a single Gabor function covers two symmetric, elliptically shaped regions in the frequency domain. Through the use of multiple filters, a broad range of spatial frequencies can be covered. Figure 15 is a frequency domain plot of the spatial frequencies covered by multiple Gabor filters. The ellipses represent contours of equal response for the example filters. Gabor filters have optimal joint resolution in the spatial and frequency domain (26, 15, 16). A minimum number of filters are needed to perform spatial frequency analysis.

The Gabor filter is both orientation-selective and pitch-selective. The output of a Gabor filter will indicate the degree a particular pitch and orientation are present within its receptive field. Multiple Gabor filters can be used to measure the orientation and pitch content at each location in a digital image. Measuring spatial frequency content at multiple spatial locations is known as spatial frequency analysis. Substantial experimental evidence suggest that some form of spatial frequency analysis is being performed in the primary visual cortex (3, 20, 30, 62, 72, 88).

3.3 Combining Spatial Frequency Information with the PCNN

Several methods of combining information are observed in the biological visual cortex. Two observed methods are summing individual attributes, and selecting attributes by magnitude. The direct convergence (summing) of different sources, registering different attributes of the visual scene, is not the predominant or preferred approach that the cortex uses to combine different sources (97). Each stage of each visual pathway contributes to perception explicitly (97). In the early stages of visual processing, neuronal processing units measure the amount of information, to which they are selective, at their location (90). Neurons that detect information to which they are selective provide greater output than those that do not. The neurons with the greatest output represent the type of information most present in a visual scene. In our visual model, this can be simulated by simply letting the filter with the greatest output at each point in the visual scene represent the type of information most present at that point. The goal in this section is to determine which filter has

the greatest output at each location in the visual scene (input image) and retain only the output of those filters as features. Filters have an inherent space/frequency trade-off. Physiologically motivated feature competition is used to reduce the effects of this trade-off.



(a) (b) (c) (d)

Figure 16 Square processed by cosine Gabor filters. (a) 3 × 3 pixel square (b) square processed with Gabor filter oriented to 0 degrees (vertical) (c) square processed with Gabor filter oriented to 45 degrees (d) square processed with Gabor filter oriented to 90 degrees

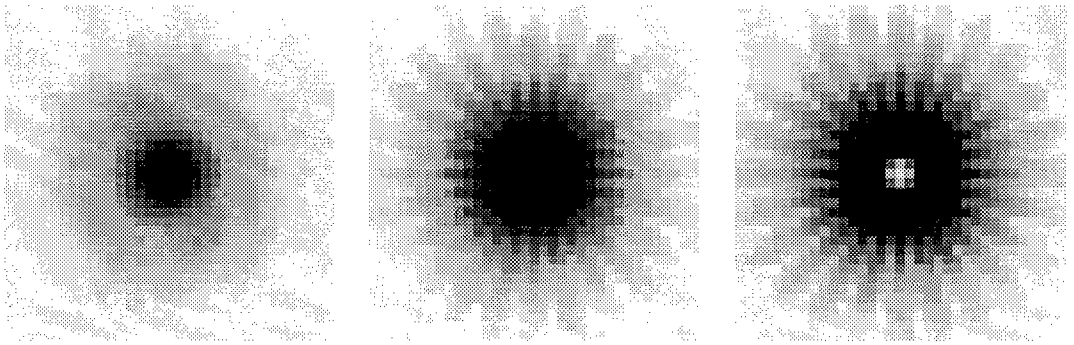
Biological evidence shows the neuronal processing units in the primary visual cortex which combine information produced by orientation-selective cells each have receptive fields that cover a small area in the visual field (97). A feature produced by these processing units is not based on a single point in the visual scene, but represents information at every point in its receptive field. The size of these receptive fields causes a degree of uncertainty as to the location of a detection within the field. The receptive field of each processing unit overlaps with the fields of other processing units (97) which adds additional uncertainty to the spatial location of objects detected in the visual scene. This spatial uncertainty is demonstrated in the following example.

In this example, the cosine Gabor filter is used to simulate the response of spatial frequency-selective cells in the visual cortex. Twelve filters are used to extract orientation information (at the filter's preferred pitch) from the 3 × 3 pixel square shown in Figure 16a. The 12 filters detect the same pitch, but differ from each other in orientation. The filters are oriented every 15 degrees which covers all multiples of 15 degrees in a 360 degree circle. Each oriented filter's impulse response is convolved with the image of the square and orientation features are determined from the filter outputs. The goal is to select the filter with the greatest output at any given spatial coordinate. The orientations of the selected filters represent the dominant orientation (at the filter's preferred

pitch) that exist at each coordinate. Figures 16b, 16c, and 16d show the output of three of the Gabor filters which are oriented at 0, 45, and 60 degrees (from vertical), respectively. Like the outputs of neuronal processing units, the filter output at any given spatial coordinate represents orientation information within a region (the filter's receptive field) about that coordinate within the input image. The filter receptive fields overlap just as the neuronal receptive fields do. These multi-pixel, overlapping receptive fields cause a degree of spatial uncertainty. This uncertainty results in each point in the square being represented by a pattern the size of the filters response (Figure 14a). The spatial uncertainty of each filter can be seen in the filter outputs shown in Figure 16.

Individually, these filter outputs give little information about the size, shape, and location of the detected object. Many existing vision models attempt to decrease these spatial uncertainties by using physiologically motivated competitive operations between filter outputs (38, 60, 25, 66, 40, 39, 12, 36, 35, 37, 13, 34). These operations include lateral inhibition and winner-take-all competitions which are demonstrated in this example. For digital simulations, these operations are typically applied on a pixel by pixel basis due to the pixel-based nature of the digital image. Since both neuronal processing units and filter units each operate on a region of pixels, pixel-based processing does not completely simulate a competition between processing units. The goal of competition is to have the unit with the greatest output suppress or over-ride the output of all other competing units. Through pixel-based processing, other filter detections cannot be fully suppressed or over-ridden.

Figure 17 shows the results of three pixel-based operations. The corresponding operations are a pixel-wise sum, a "winner-take-all" operation, and a "winner-take-all with lateral inhibition" operation. The later two are competitive operations. Figure 17a shows the results of summing the pixels of the filter outputs shown in Figure 16 (the output of 12 filters are summed). To perform the summing, the pixel intensities of all filter outputs at a single (x, y) location are summed to form a single value for that (x, y) location. Since no dominant orientation is determined at each coordinate,



(a) (b) (c)

Figure 17 Various methods of combining Gabor filter outputs. Filters are oriented at every 15 degrees for a total of 12 filters. (a) Filter outputs combined by summing all filter outputs at each spatial location. (b) Filter outputs combined by keeping only the max filter output at each spatial location. (c) Filter outputs combined by applying lateral inhibition between pixels in each orientations then keeping only the max intensity pixel at each spatial location.

this pixel-wise operation loses all orientation information about the square. It also suffers from spatial uncertainty, since the object's boundaries cannot be determined from the output. Figure 17b shows the 12 filters combined with a pixel-wise Max (winner-take-all) operation. The Max operator retains the maximum filter intensity at each (x, y) location and discards all other filter intensities at that location. The Max operator retains orientation information by selecting the filter with the greatest output, but the pixel-wise application method still leaves much spatial uncertainty remaining. The spatial uncertainty stems from the pixel-wise operation's inability to discard the entire output of non-selected filters. For this square, the outputs of the filters oriented at 45 and 135 degrees had greater energy than the outputs of any of the other orientations. If our goal were met, only these two orientations should have been selected and all orientations should have been suppressed or discarded. The pixel-wise Max operation could not discard the entire filter output, only the individual points at which the filters' receptive fields overlap. A well-known vision model, the Grossberg boundary contour system, performs pixels-wise lateral inhibition across filters, and then performs a pixel-wise Max operation (38). Figure 17c shows the results of this process. This method retains orientation information, but suffers from spatial uncertainty. As with the Max

operation, the pixel-wise lateral inhibition can only suppress pixels of non-selected orientations at points where the filter receptive fields overlap.

These pixel-based approaches can be improved by grouping all pixels in the output of a simulated processing unit to form a single entity (object). The same physiologically motivated competitions used earlier can be performed between objects instead of pixels. Competitions in the biological vision system are performed between neuronal processing units and not the individual locations within their receptive field (97). For this reason, competition between objects simulates physiology with greater fidelity than competition between pixels.

(a) (b) (c) (d)

Figure 18 Square in Figure 16 processed by cosine Gabor filters then segmented with PCNN. (a) PCNN object segmented from 0 degrees oriented filter output (b) PCNN object segmented from 45 degrees oriented filter output (c) PCNN object segmented from 90 degrees oriented filter output (d) PCNN objects combined by keeping only the max intensity object at each spatial location

This object-based approach to feature extraction can be implemented in many ways. The segmentation capability of the PCNN provides an efficient and effective physiologically motivated method for both grouping pixels into objects, and performing competition between objects. The temporal synchronization property of the PCNN is used to group all pixels detected by individual oriented filters into objects that can be treated as single entities. Figures 18a, 18b, and 18c show the Gabor filter outputs after the PCNN has segmented them into objects. Note the majority of each filter output has been grouped to contain a single gray level. This gray level coding is used only for display purposes. The object could have easily been coded with a unique value or object number. A competitive operation can be used to determine which object has the greatest magnitude, and the remaining objects can be easily suppressed in their entirety.

In a pulse once scenario, a second PCNN with feeding inputs connected to the pulsed outputs of several of these object forming PCNNs would exhibit a behavior identical to a Max operator. Each neuron in the second PCNN pulses in response to the first pulse it receives, then remains dormant for the remaining period of execution. As stated previously, the earlier a neuron pulses, the greater output frequency it would produce. Each neuron in the second PCNN latches the highest frequency signal received on its feeding inputs, thus simulating the Max operation. By connecting the input of each neuron in this Max PCNN to the output of the neurons (at the same coordinate) in the object forming PCNN, a competition between neurons is formed. As previously stated, the PCNN initially pulses at the brightest point in an object. This pulse causes a linking wave that synchronizes all neighboring neurons with like inputs. The initial pulse is the seed the PCNN uses to form individual objects. If the object forming PCNNs are slightly modified to transmit only this seed pulse to the Max PCNN, each entire object is now represented by a single point. If a seed point is the earliest (highest pulse frequency) to reach the Max PCNN, the Max PCNN pulses producing a linking wave which replicates the object in the Max PCNN. If a seed pulse arrives at a particular neuron in the Max PCNN that has already pulsed, no new pulse is generated. Since no pulse is generated, no linking wave is produced and the object is not replicated in the Max PCNN. This lack of replication effectively suppresses all competing objects (in their entirety) once one object has been selected as having the greatest output.

Applying this PCNN-based competitive process to the objects in Figures 18a, 18b, and 18c produces the output shown in Figure 18d. The object-based competition retains orientation information and reduces the spatial uncertainty present in the original filters. This process achieves the goal of selecting the filter with the greatest response to the object and suppressing all other competing filter outputs.

This feature extraction process is shown in block diagram form in Figure 19. The first PCNN in the process segments the filter outputs into objects. The intensity of each object is directly

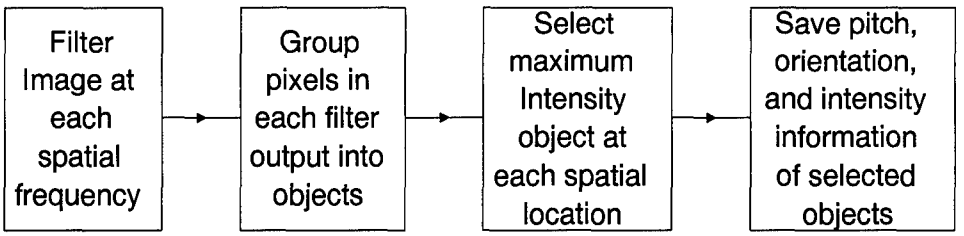


Figure 19 Functional block diagram of PCNN feature extraction process.

proportional to the total energy in the pixels combined to form the object. The second PCNN selects the maximum intensity object at each spatial coordinate which gives the pitch, orientation, and intensity of the selected objects. Using the first PCNN to group filter outputs into objects, and the second PCNN to pick the maximum valued object, forms a PCNN-based visual feature extraction network.

All Gabor filters used in this example detect the same pitch, but differed in preferred orientation. Attempting to select the filter with the greatest response at a particular coordinate from a group of filters that differ in pitch poses the same difficulties encountered when selecting from filters of various orientations. The PCNN object based filter selection techniques can be used to select between filters of different pitch in the same way it selects between filters of different orientations. Extending the concept of PCNN feature extraction to include filters that differ in pitch selectivity produces the PCNN feature extraction network shown in Figure 20. This network segments filtered images into objects, selects the maximum intensity object at each spatial coordinate, and records the pitch, orientation, and intensity of the selected objects.

The functionality of the network is independent of the characteristics of the chosen filters. Wavelet filters, Difference-of-Gaussians (DoG) filters, or any other spatial frequency selective filter could be substituted for, or combined with the Gabor filters. The network will perform a spatial frequency analysis using any of these filters. As previously stated, it is hypothesized that neuronal processing units are best described as filters that are selective along multiple stimulus directions (90). This network can extract features using any filter that is selective along multiple

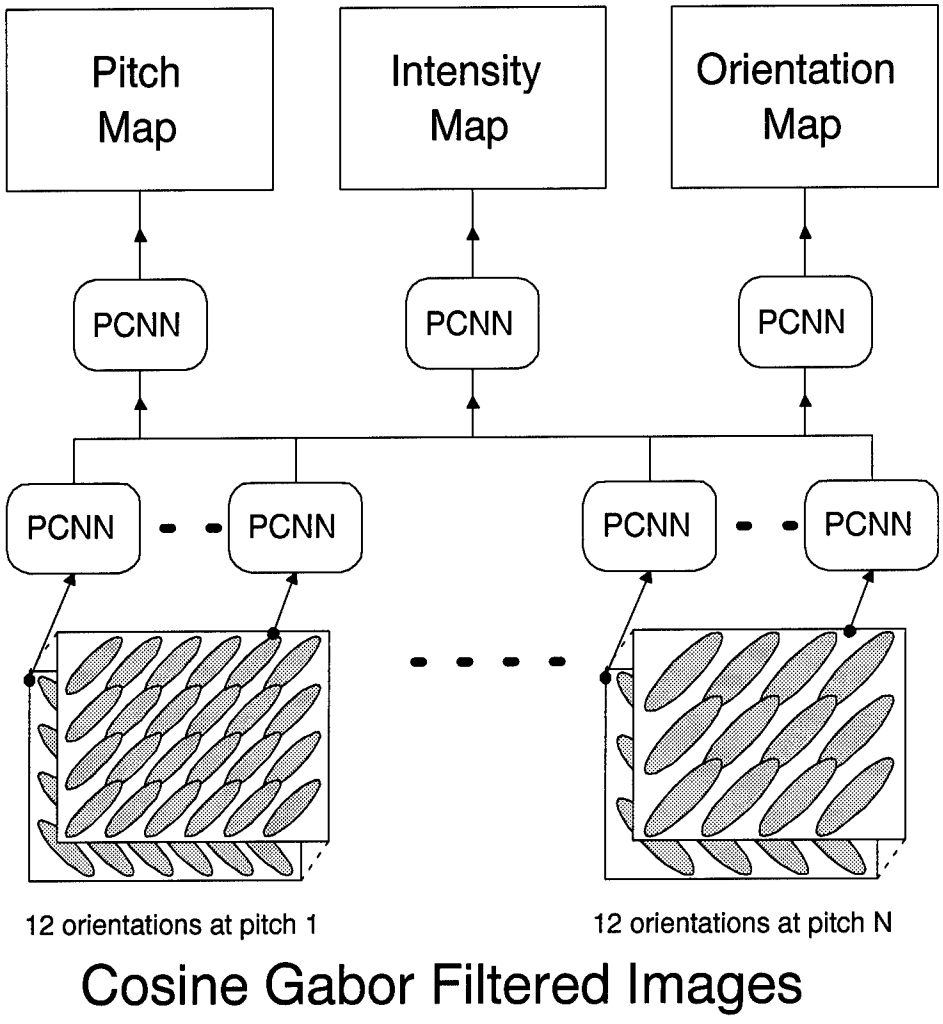


Figure 20 PCNN feature extraction network.

stimulus directions. The network can extract motion features using spatio-temporal filters and can extract color features using spatio-wavelength filters. This allows easy extension of the network for analysis of color imagery and sequential image sets containing motion (video).

The accuracy of the extracted features is driven by the number and characteristics of chosen filters. For example, a 600 filter network would provide better feature resolution than a 60 filter network. A network constructed with non-orientation-selective filters (e.g., DoG filters) would produce no orientation features.

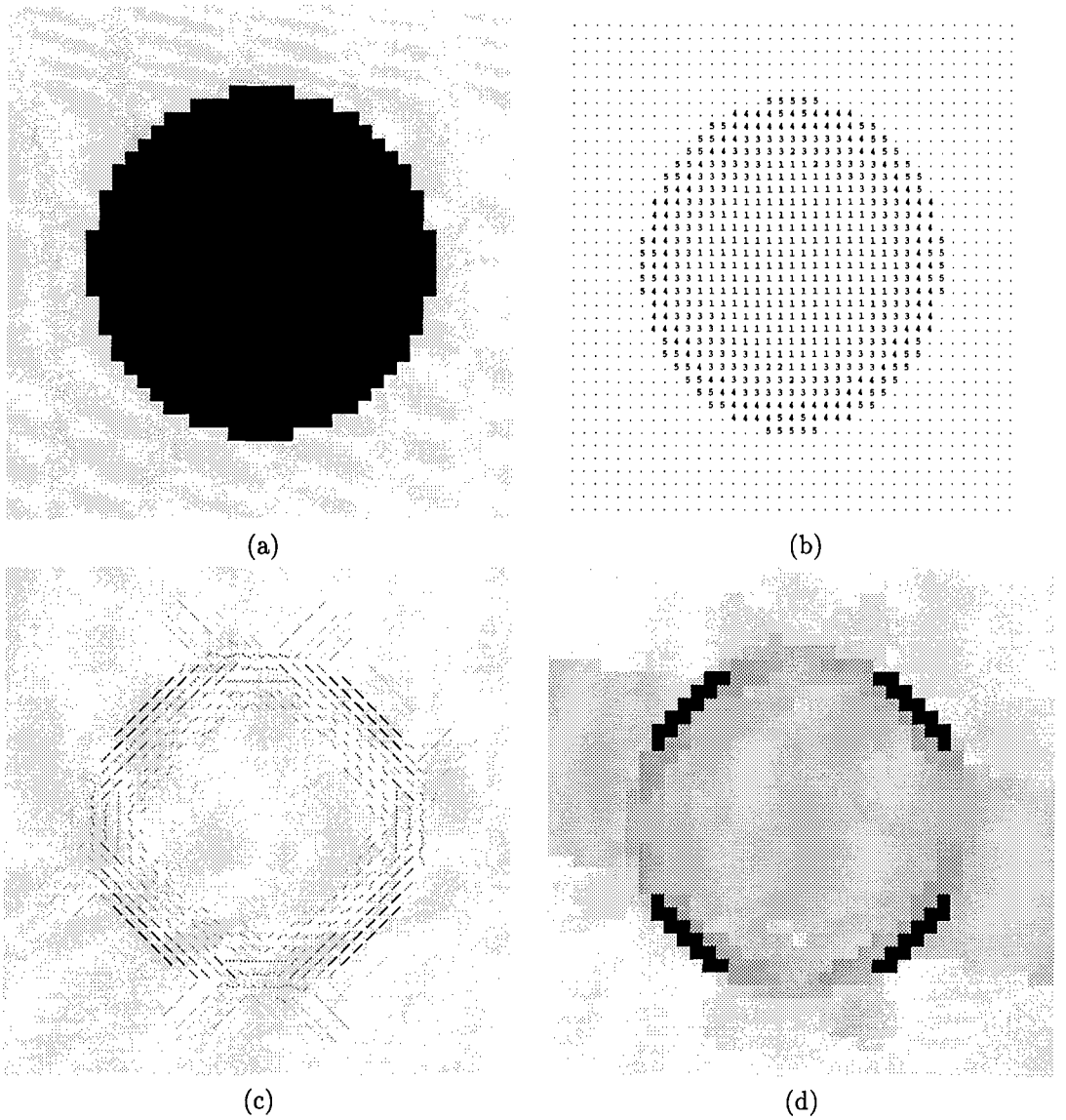


Figure 21 Simulated visual features extracted by the PCNN feature network. (a) Original image (b) pitch feature map (c) orientation feature map (d) intensity feature map

The PCNN feature extraction network simulates the spatial frequency analysis which experimental evidence suggest is being performed in some form in the primary visual cortex (3, 20, 30, 62, 72, 88). The pitch, orientation, and intensity selected at each location is the simulated visual feature for that location. Figure 21a shows a circle, and Figures 21b, 21c, and 21d show the features extracted from the circle by the PCNN feature extraction process. The image was filtered using 60

cosine Gabor filters each centered at the spatial frequencies shown in Figure 15 (five pitches at 12 orientations each). Figure 21b shows the dominant pitch present at each point in the circle. The numbers denote the pitch of the selected filter (higher numbers indicating higher frequencies). The five filter groups are an octave apart in pitch. A number 5 denotes a pitch of 128 cycles per image, 4 denotes 64 cycles per image, 3 denotes 32 cycles per image, 2 denotes 16 cycles per image, and 1 denotes 8 cycles per image. The frequencies are displayed in this format to allow each pitch to be represented by a single digit. Figure 21c shows the dominant orientation present at each point in the circle. The orientation map has been multiplied by the intensity map for display purposes. Darkness of line segments denote the relative presence of the orientation (ie, locations with light line segments are not as strongly oriented as locations with dark line segments). Figure 21d shows the intensity at which the dominant pitch and orientation are present at each point. In other words, this map gives the strength with which the selected filter responded.

3.5 Simulating Focus of Attention with the PCNN

To detect the presence of a desired object within a visual scene, an object detection algorithm can suppress all objects that do not have features matching the desired object. Only objects that have features resembling the desired object will remain. Alternately, objects that have features resembling the desired object can be enhanced. The two methods are equivalent except for scaling. The process of enhancing desired objects (or features) can be called a focus of attention. Additional attention is focused on the desired object or features. Focus of attention can easily be added to the PCNN feature extraction network by adding a positive bias to desired features.

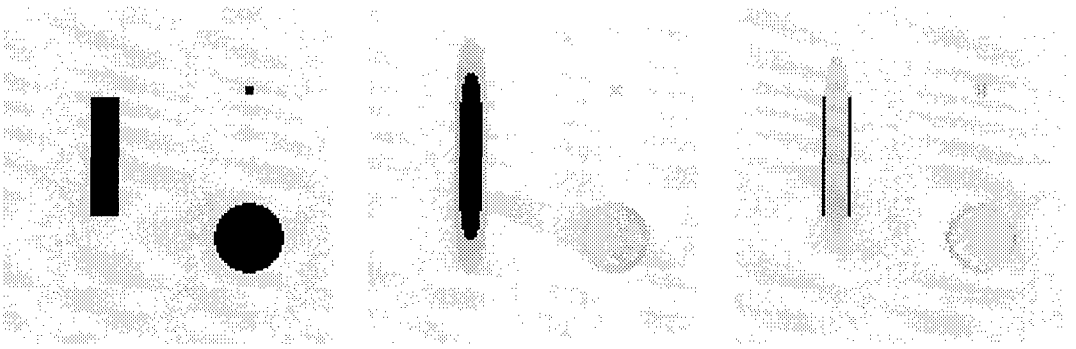
The biological principle of state dependent modulation can be used as a mechanism for focusing attention on features of a desired object. In the biological vision system, state dependent modulation signals increase a neuron's response to its input. A signal of this type can be used to increase neuronal response to desired features which in turn elevates the overall visual response to

a desired object. This elevated response facilitates detecting and isolating a particular object in a visual scene.

The principle of state dependent modulation can be easily applied to the feature extraction system described in this chapter. Extracted features are the spatial frequency information derived from the outputs of multiple filters. To isolate a desired object in an image, a positive bias can be added to the output of filters which are selective to the desired features. This bias causes the desired object to be enhanced relative to the rest of the visual scene. All desired features will be the brightest features in the processed visual scene. To increase the signal-to-noise ratio, the bias signal can be applied by multiplying instead of adding (44). Lower intensity noise signals are not increased as much as the higher intensity filter detection signals. Multiplying one signal by another is called modulation. The bias signals multiplied against select filter outputs is a state dependent modulation signal. These modulatory signals focus attention on desired objects in the visual scene. To shift the focus of attention from one object to another, simply shift the state dependent modulation signals to different filters. The focus of attention moves to objects with different characteristics.

As discussed in Section 2.1, the linking inputs of the PCNN modulate the feeding inputs. State dependent modulation signals are applied to the filter outputs by applying the bias signal to the linking inputs of each neuron in the PCNNs that process the output. This increases the output frequencies produced by the PCNN which elevates its output above other non-biased PCNNs. The features produced by the biased PCNNs will have the greatest magnitude in the output of the PCNN feature selection system. This modulation process simulates the state dependent modulation observed in the biological vision system. Through this mechanism a focus of attention can be applied to desired objects, thus forming an object detection system.

Figure 22 shows an example of focus of attention. The image in Figure 22a contains three objects; a tall rectangle, a circle, and a small square. The goal of this example is to detect the tall



(a) (b) (c)

Figure 22 Example of focus of attention using state dependent modulation. Goal is to enhance long vertical edges to distinguish tall rectangle from other objects. (a) Original image (b) intensity features produced by PCNN feature extraction network (c) intensity features produced with state dependent modulation signal of 3 applied to PCNN that processes highest frequency vertically oriented features.

rectangle by focusing attention on one of its distinguishing features. The long vertical edges will be used as the distinguishing feature since the rectangle is the only object in the scene that contains them. Figure 22b shows the intensity features extracted from the image when no state dependent modulation signals are present. A state dependent modulation signal, of magnitude equal to 3, is applied to the particular PCNN which processes the highest pitch vertically oriented filtered image. With this focus of attention, the PCNN feature extraction network produces the intensity features shown in Figure 22c. The long vertical edges of the tall rectangle are the brightest features present. This added intensity can be used to easily detect and isolate the desired object in the visual scene.

3.6 Summary

PCNNs and Gabor filters were used to simulate the biological feature extraction performed in the primary visual cortex. The feature extraction model uses Gabor filters to simulate the biological orientation-selective vision cells and the PCNN to simulate the cells that compare and select visual features produced by these orientation selective cells. The resulting features describe the pitch, orientation, and intensity that exist at each location in an input image. This feature extraction network models the spatial frequency analysis shown to exist in the primary visual cortex.

Some degree of spatial uncertainty is inherent in all spatial frequency filters. pixel-based versions of physiologically motivated techniques for reducing this uncertainty have been demonstrated to show the inadequacies of pixel-based methods. The temporal synchronization property and refractory period of the PCNN are used to provide a superior object-based alternative to pixel-based methods. Using these physiologically motivated principles, the PCNN forms objects from the filter outputs, and compares these objects to determine the features that exist at each spatial location. This object-based approach does not produce the feature artifacts that plague pixel-based approaches. Through examples, the features produced by the PCNN feature extraction network are compared to features produced by several pixel-based methods. The PCNN-based system produces features that have greater spatial precision and contain less artifacts than the features produced by the pixel-based techniques. The physiologically motivated principle of state dependent modulation is used to add a focus of attention capability to the PCNN feature extraction network, forming a simple object detection system. Through a simple example, this focus of attention capability is used to detect a desired object within a visual scene containing several objects.

The strength of this feature extraction network lies in its flexibility. Simple modifications have been presented that can extend the model's capabilities to perform spatio temporal (motion) and spatial wavelength (color) analysis. With these extended capabilities, the feature extraction model can simulate visual processing of all known basic information types (luminance, wavelength, direction, and orientation) processed by neuronal processing units in the early stages of the primate vision system (90, 98, 97, 7, 19, 14). Cascading this model to simulate observed multi-layer hierarchical vision processing can produce the higher order moments of the basic information types such as gradient information, texture, and acceleration (89). This set of features provides a sufficient basis for nearly any type of visual object detection/recognition goal. The extended model can provide an effective, flexible, and extensible feature extraction stage for nearly any object recognition system.

Object detection systems, similar in principles to the one developed in this chapter, are used in the next chapter to simulate biological information fusion. The physiologically motivated principles of temporal synchronization and state dependent modulation are used to combine the outputs of several object detection systems to increase object detection accuracy. This information fusion system is demonstrated on real-world images with promising results.

IV. Information Fusion for Object Detection

4.1 Overview

Digital image processing is being investigated for object detection in applications such as breast cancer detection and automatic target recognition (29, 68, 85, 58, 59, 78, 17, 65, 71). Image processing is used to reduce unwanted information from an image with the hope that the improved signal-to-noise ratio will allow a pattern recognition process to detect and possibly identify the desired object. In general, no single image processing technique can be selective to all patterns for a given object, and still perform well at removing the many possible variations of unwanted information. Often, several techniques are used and the results are combined.

As previously mentioned, many current theories propose that neuronal pulses synchronize to combine visual features into visual objects (32, 23, 67, 92). In this chapter, these theories are used to design a PCNN-based image fusion network that segments a visual scene, combines features to form objects, and isolates desired objects from the rest of the image. This PCNN fusion network combines the output of individual detection techniques in a physiologically motivated fashion for the purpose of improved object detection. Observed biological phenomenon such as temporal synchronization and state dependent modulation are applied to combine the information and focus attention on a desired object. The role that these biological phenomena perform in information fusion and in the image fusion network is discussed. Through a combination of image segmentation, information fusion, and attention focus, an object detection property emerges from the PCNN fusion network. Actual infrared and mammographic images are used to demonstrate the object detection accuracy of the network (6).

4.2 The PCNN Fusion Network

To perform object detection, the PCNN fusion network takes an original and filtered versions of a gray-scale image and outputs a single image in which the desired objects are the brightest

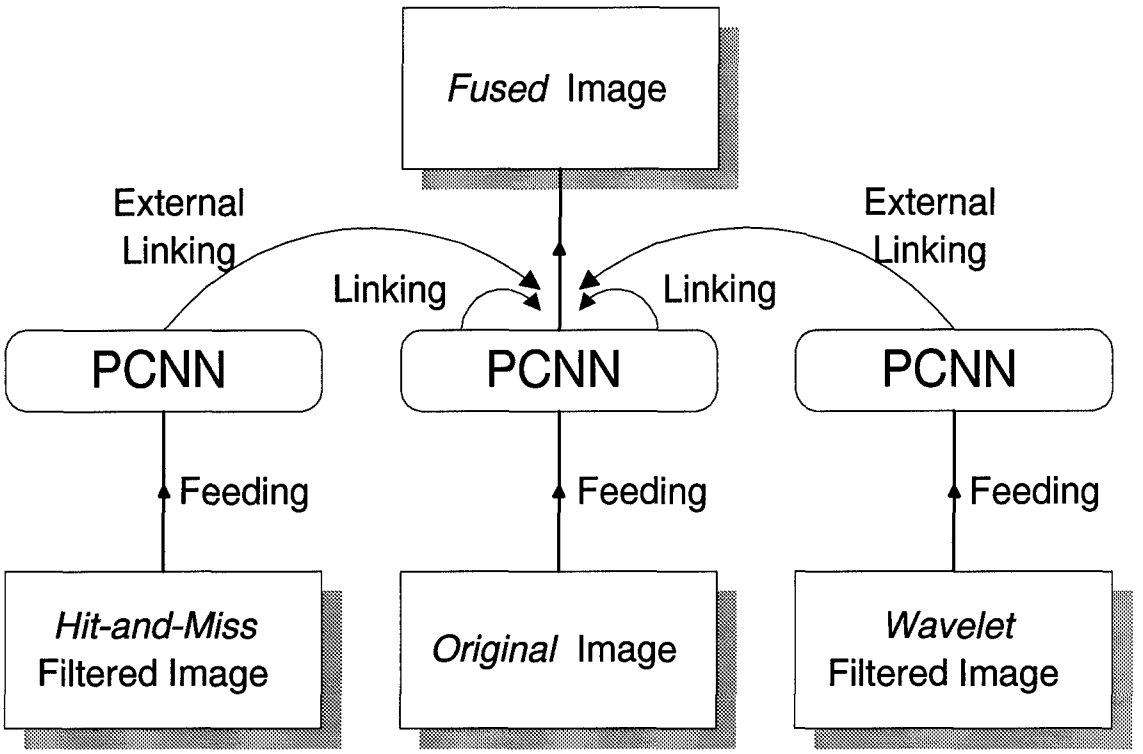


Figure 23 PCNN fusion architecture used to fuse both breast cancer and FLIR images.

objects and thus easily detected. Gray-scale outputs of object detection techniques are used as inputs to the fusion network. These gray-scale images are used to simulate the feature maps produced in the previous chapter. The filtering process simulates the feature extraction process. Each filter is tuned to be selective to a particular characteristic of a desired object which simulates focus of attention.

Figure 23 shows the PCNN network used to fuse the original and filtered images. When applied to the mammograms, the image processing filters are tuned to be selective to microcalcifications which can be an early indication of cancerous growth (68). For the FLIR images, the filters are tuned for selectivity to features of a SCUD mobile missile launcher. Since these filters are selective to a particular object, the outputs can be used as state dependent modulation signals where the current state of attention is focused on detecting objects that resemble the target object.

Each PCNN in Figure 23 has one neuron per input image pixel. The average output pulse rate of each neuron in the center PCNN is used as a brightness value for the pixels in the output image. Each neuron is allowed to pulse only once during execution (pulse-once scenario), therefore the period (timestep) of the output pulse is used to calculate an average output pulse rate for each neuron. The neurons within the PCNN are arranged as a single two dimensional layer network with lateral linking. Figure 2 (page 13) shows the feeding and linking connections of a single neuron within the PCNN. As used in this chapter, every neuron receives linking inputs from all neighboring neurons within a radius of 3 (Figure 2 shows a linking radius of 1). Each neuron receives feeding inputs which are the intensity of the corresponding pixels in the input image. The pulse-based linking mechanisms of the PCNN use temporal synchronization to segment the original image. The outer PCNNs provide state dependent modulation signals used to focus attention on segments of interest.

Figure 24 shows the inputs and output of the fusion process when used on a small portion of a mammogram which contains microcalcifications. The average pulse rate of each output neuron is used as a brightness value for the pixels in the output image. Figures 24a, 24b, and 24c are the images used as input to the fusion network. The fusion results are shown in Figure 24d. A threshold has been applied to remove the background and lower intensity segments. The segments that remain are the desired objects.

4.3 Pulse Coupling Performs Temporal Synchronization

PCNN pulse synchronization is discussed in Section 2.1.6. Pulse synchronization causes neurons with similar inputs to form a synchronously firing group. This grouping results in segmentation of the input image. Segmentation allows the PCNN fusion network to identify and remove unwanted objects based on size, shape, and intensity.

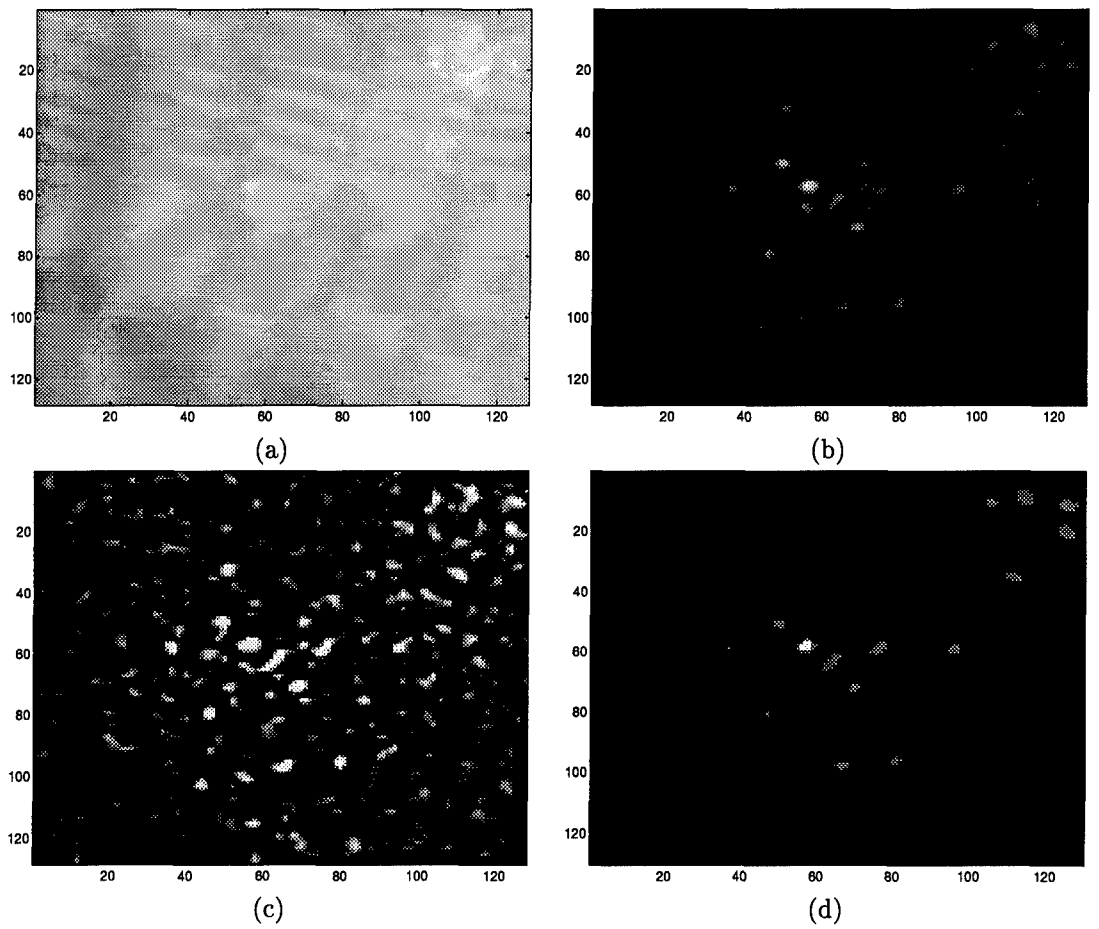


Figure 24 128-by-128 pixel region containing microcalcifications (segmented from a 1024-by-2048 pixel mammogram). (a) Original image (b) hit-and-miss filtered image (c) wavelet filtered image (d) PCNN fused image after a threshold has been applied.

4.4 State Dependent Modulation in the PCNN Fusion Network

The PCNN fusion network uses the principle of state dependent modulation to focus attention on objects that best fit the criteria of a desired object. By using the relative presence of a desired feature as a state dependent modulation signal, the network's response to the desired object is elevated. This elevated response facilitates detection and isolation of a particular object in a visual scene.

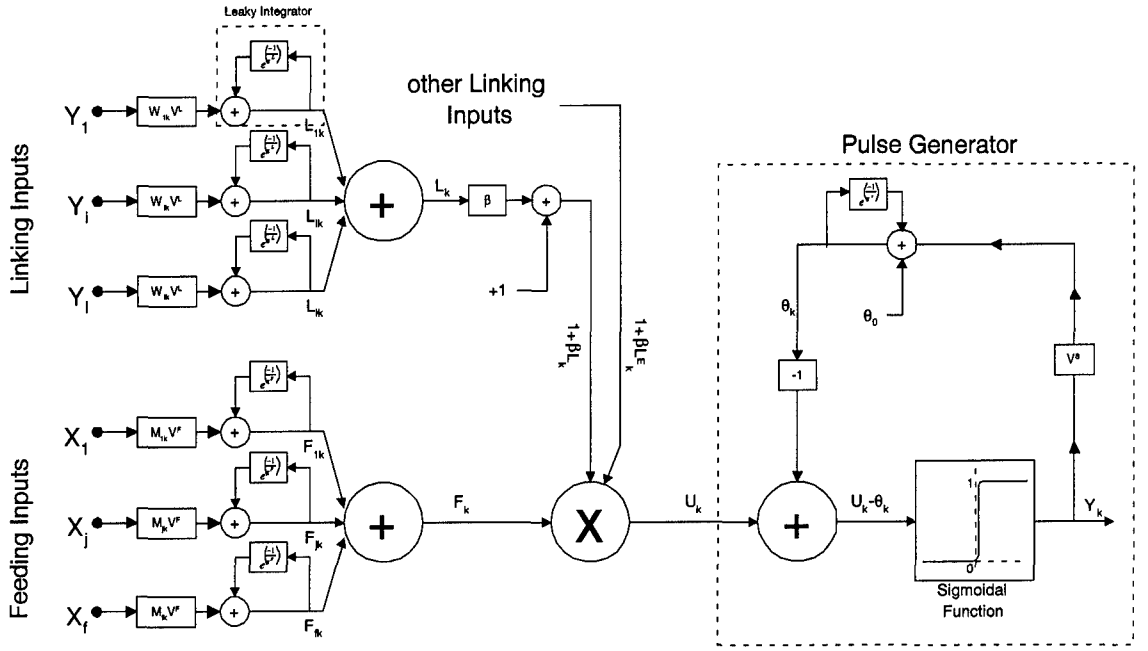


Figure 25 The Eckhorn artificial neuron used within the PCNN.

As shown in Figure 25, the total input to a PCNN neuron (U) can be described by the equation

$$U = F(1 + \beta L)(1 + \beta ExtL) \tag{10}$$

where $ExtL$ is the value of total linking inputs from sources external to the PCNN (possibly other PCNNs). The signal U feeds directly into the pulse generator section of the PCNN which produces the output pulse train. The output frequency of pulses produced by the pulse generator is

$$f = \frac{1}{\lceil -\tau_s \ln\left(\frac{U}{V^S}\right) \rceil} \tag{11}$$

which is the reciprocal of the output period shown in Equation (4). From Equation (10) it can be seen that the linking inputs of the PCNN modulate the feeding inputs. This is the modulatory mechanism used to simulate the state dependent modulation. Without linking inputs, U would equal F and the feeding input would drive the pulse generator section. A positive linking input ($L > 0$) would increase the value of U which would increase the frequency of the output pulse train

(Equation (11)). If outputs of filters which are selective to features of a desired target are used as linking inputs, then neurons connected to image areas that resemble the desired target would have greater linking inputs than those that do not. The filter outputs represent the state dependent modulation signals when the current state of the PCNN is a focus of attention on the desired target. The neurons with inputs that best match the desired target would have the greatest modulatory input, thus having the highest frequency output. This increased output effectively separates the neurons from the rest of the image.

For the PCNN fusion network, this modulatory mechanism provides a method of associating filtered features with segments in the original image. It also provides a focus of attention to isolate the segment. Segments with a greater number of desired features present will be more active than other segments; therefore the most active segments are those that fulfill more of the target criteria. These segments are easily separable from the rest of the image.

4.5 How Information is Fused

The cornerstone of the PCNN fusion network is the segmentation performed by pulse synchronization. This temporal synchronization groups the image pixels into individual, disjoint segmented regions (objects) that pulse at different frequencies (46, 74). The parameters of the PCNN are manually set to segment image regions fitting the desired object's size and brightness characteristics into single objects. Since the PCNN segments on brightness boundaries, the PCNN parameter values used in the segmentation process are image dependent (74). PCNN segmentation is sensitive to image contrast, thus some images sets may require preprocessing to ensure all images within the set have similar contrast. Histogram equalization has performed satisfactorily as a preprocessing step for many of the images in this research. The quality of the information fusion process is highly dependent upon the quality of the image segmentation. Chapter V presents an adaptive

PCNN that can be used to determine the PCNN parameter values necessary to achieve a desired segmentation result.

The fusion process exploits the fact that the segmentation step has grouped the input image into objects. Since the features produced by a feature extraction process may be spatially disjoint, a method is needed to associate features belonging to a single object to that object. The objects produced by the PCNN segmentation provide a single region of space to which disjoint features can be mapped. These method allows dissimilar and possibly spatially disjoint features such as brightness, edges, and gradients to be associated with individual objects. Through this feature association, several dissimilar features of an object are fused into a single representation of the object.

In the PCNN fusion network, the original image is used as a basis for object segmentation, and the filtered versions of the original image are used as the dissimilar features. The filters used in the fusion process are tuned to be selective to particular features of the desired object. Each filtered image, produced by convolving the impulse response of a tuned filter with the input image, represents image features with a focus of attention on a particular characteristic of the desired object. The two outer PCNNs shown in Figure 23 convert the filtered images into pulsed signals for use as state dependent modulation signals. These pulsed signals are linked to the original image using the center PCNN's linking inputs. These linking connections are arranged such that each neuron in the outer PCNNs provides a linking signal to the neuron in the center PCNN that occupies the same relative spatial location.

The purpose of these linking signals is to link (fuse) each individual feature into its associated object. These signals are linked by modulating the center PCNN's neuronal response to the object of interest. The modulatory signals received by each neuron within an object increases the magnitude of the total input signal U which increases the pulsing frequency of each neuron. Local linking connections within the center pcnn cause the neurons to fire in synchrony as a single object (46, 45).

This is the method by which dissimilar and possibly disjoint features are associated with individual objects. This association process fuses information from separate images into a single image.

The object detection property of the PCNN fusion system is inherent to the association process. The strength of the total modulatory signal present within each object is directly proportional to the number of desired features present within the object and degree to which the features are present. Objects that contain a greater number of desired features receive a larger modulatory signal than objects that do not. The greater the modulatory input an object receives, the higher the pulsing rate of the neurons within the object. The pixels within the original image that best fulfill the selective criteria of the filters will be represented by the fastest pulsing objects in the output. Since the value of each pixel in the output of the PCNN fusion network is the pulsing frequency of the corresponding neuron in the center PCNN, objects with higher pulsing rates are represented as brighter pixels. The brightness of the output pixels can be used to effectively separate the desired objects from other objects and the image background. Brightness thresholding of the fused output image can be used to remove background objects leaving only objects that whose features resemble the desired object. This is the object detection property inherent to the PCNN fusion network.

4.6 Object Detection Results Using X-Ray and FLIR Images

The following example demonstrates the object detection capability of the PCNN fusion network. The network is used to fuse information from two independent object detection systems to produce a single output that has fewer false alarms while still detecting the desired object. The individual object detection systems used to generate the inputs to the fusion system are actual published detection systems. Two particular object detection systems are chosen to provide visual features because both have been used successfully to detect breast cancer in mammograms (68, 17, 65, 71) and SCUD missile launchers in FLIR images (78). One of the systems is based on morphological (hit-and-miss) processing, and the other is based on Difference-of-Gaussians (DoG)

filtering. The band pass filtering performed in these detection systems are used to extract size and spatial frequency components from digital images. The filtered components serve as individual features which are fused by the PCNN fusion network into a single image which combines and yet exploits the selectivity of each individual filter.

The two detection systems serve the same function as the PCNN feature extraction network presented in the previous chapter. These object detection systems can be viewed as a feature extraction system that has a focus of attention for a particular type of object (the target). This focus of attention is created by selecting the band-pass filter characteristics that best detect the desired object. The features produced by the two systems are a subset of the features produced by the more general feature extraction network. A frequency output is not produced because the detection systems are tuned to a specific frequency range. Since detection of the target is desired at any orientation, non-oriented filters are used. This means an orientation output is also not needed or produced. This leaves the intensity output as the remaining feature. The outputs of the detection systems are very similar to the intensity output of the feature extraction network. Each output contains a measure of the energy in the bandpass filters bandwidth at each point in the original image.

Figures 26a, 26b, and 26c show example inputs to the image fusion network. Figure 26d shows the output produced by the network. Figure 26a is an actual FLIR image produced by an aircraft imaging system. The image contains a SCUD mobile missile launcher, two support vehicles, and four flash pods. Figures 26b and 26c are the outputs of the two object detection systems when using Figure 26a as an input.

In this experiment, 100 FLIR images were used to calibrate and test the object detection capability of the PCNN fusion network. The goal of the experiment is to detect the SCUD launcher while minimizing the number of false alarms. Fifty images were used to calibrate the PCNN weights, linking radius, β and threshold parameters. Each image contained a single SCUD mobile missile

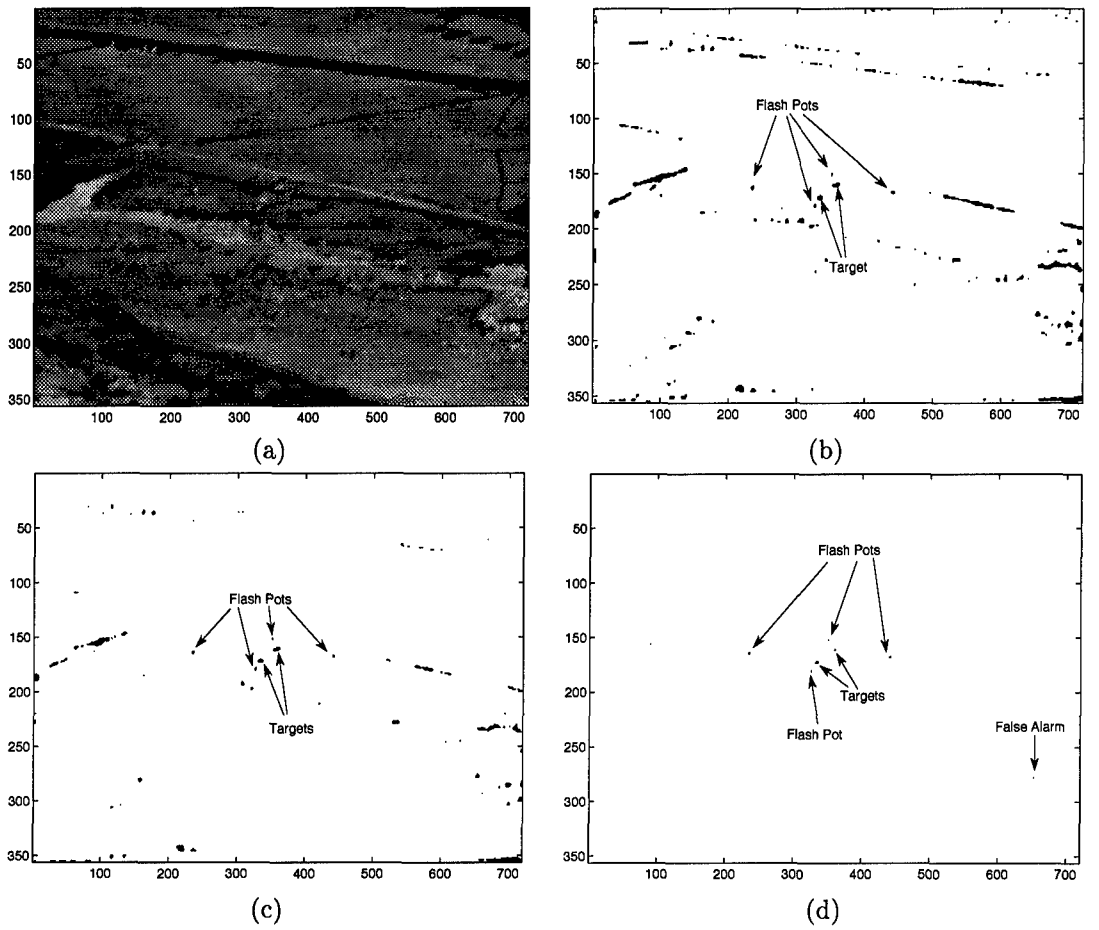


Figure 26 Input and output images of a mobile SCUD launcher and flash pods. (a) Original FLIR Image (b) DoG filtered image (c)morphological (Hit/Miss) filtered image, (d) PCNN fusion network output image

launcher, a truck, a van, and four surrounding flash pods to mark the target location. The flash pods function as a guide for the photographer and are not used by the detection algorithms. After PCNN calibration, the object detection capability of the fusion network was tested on the remaining 50 FLIR images. Since the output of an object detection system is often processed by a pattern recognition engine to obtain additional accuracy, a large false alarm rate is preferable to a missed target. For this reason, all filters were tuned conservatively to ensure SCUD detection. Any detected object other than the SCUD, truck, van, pods, and image edge effects were considered false targets. Detection of the truck, van, and flash pods was considered optional.

Table 4 Mobile SCUD launcher detection results using the PCNN fusion network

Image Number	Number of False Alarms (with 100 percent target detection)		
	Hit/Miss algorithm	DoG algorithm	PCNN network
1	7	12	0
2	5	22	0
3	2	16	0
4	28	17	1
5	40	28	2
6	9	7	3
7	1	8	1
8	1	26	0
9	1	30	0
10	3	29	0
11	7	27	0
12	3	21	0
average	8.2	20.3	0.6

Table 4 presents the detection accuracy achieved by each method. Due to space considerations, only 12 images are shown. The accuracy achieved on these 12 images is representative of the average accuracy achieved over the entire image set. The images shown in Figure 26 are example images taken from this set of results. For every image inside the desired target detection range (ranges of interest from a munitions release perspective), the selective filters and the PCNN fusion network detected the SCUD mobile missile launcher. As can be seen in Figures 26b and 26c, conservative tuning can cause the selective filter routines to produce a large number of false alarms. The Hit/Miss filter algorithm averaged 8.2 false targets per image, the DoG filter algorithm averaged 20.3 false targets per image, and PCNN network averaged 0.6 false alarms per image. When compared to the best filter accuracy, the PCNN network removed 93 percent of the false alarms without removing any true detections. The accuracy produced by the PCNN network also exceeds the accuracy produced by ANDing the filter outputs.

In the second test, the algorithms were used to detect microcalcifications in mammograms. Microcalcification density is often used by computer aided diagnosis (CADx) systems for early detection of cancerous breast regions (29). Microcalcifications are present in healthy tissue, but a high density (5+ per square centimeter) can be an early indication of cancer. In this test, the

selective filters were tuned to detect radiologist identified microcalcifications. The goal of the test was to maximize the detection of identified microcalcifications while minimizing the number of other detections. All detected objects that did not represent an identified microcalcification were considered false targets. The identified microcalcifications were visually detectable, but others may exist. Since this test does not attempt to detect all microcalcifications, but only those identified by radiologist, the resulting accuracy should not be directly compared to other cancer detection algorithms. The purpose of the test is to demonstrate information fusion by a PCNN.

Table 5 Detection results of microcalcifications in mammograms using the PCNN fusion network

Image Number	Number of Calcs Found/Number of False Alarms		
	Hit/Miss algorithm	DoG algorithm	PCNN network
1	15/15	21/27	18/15
2	31/15	41/26	38/12
3	20/17	24/19	21/8
4	32/7	49/18	15/0
5	25/29	32/57	26/20
6	5/41	7/72	1/3
7	3/24	3/34	3/28
8	4/26	5/22	4/11
9	6/16	7/29	6/21
10	2/9	4/23	0/0
11	9/14	9/19	8/2
12	10/11	10/7	10/1
Average Ratio	0.76	0.60	1.24

Thirty 256 x 256 pixel regions segmented from full breast mammograms were used to test microcalcification detection. Eighteen of the regions were used to calibrate the PCNN network and the filter algorithms, and the remaining 12 were used to test the detection accuracy. Table 5 presents the detection accuracy achieved by each algorithm. Since the PCNN network fuses the results of the selective filters, no additional true detections were expected or achieved. The results do show that the number of false detections were significantly reduced with only a small reduction in true detections. The Hit/Miss algorithm averaged 1.3 false detections for each true detection and the DoG algorithm averaged 1.7 false detections per true one. The PCNN network reduced these ratios to 0.8 false detections per true detection. When compared to the best filter result, the

PCNN network removed 46 percent of the false detections while removing only 7 percent of the true detections.

The fusion network provided a greater accuracy increase on the FLIR images than on the mammogram images. The network reduced the false alarm rate from 8.2 to 0.6 false alarms per image in the FLIR images and from 1.3 to 0.8 false detections per true detections in the mammograms. In the fusion process, the PCNN network does not add true detections to the output, but instead removes false detections. Since the FLIR images contained many objects such as trees and roads that were larger than the target, the PCNN could easily segment and remove the large objects. Because the mammograms contained few large objects with consistent brightness and boundaries, the PCNN segmented the image into many small objects which prevented any significant object removal based on size. The majority of the information removal was performed by the state dependent modulation. These results imply the PCNN fusion network is better suited for processing images which contain structures that differ in size from the targets. The PCNN was able to map many false detections, such as road and forest edges, into the larger original object and subsequently remove the false detections. The tests have shown the PCNN network is suitable for removing false detections from conservatively tuned filter outputs while preserving a majority of the true detections. The network removed 93 percent of the false detections without removing any true detections in the FLIR images and removed 46 percent of the false detections while removing only 7 percent of the true detections in the mammograms.

4.7 Summary

The first PCNN-based fusion network has been developed using the primate vision processing principles of temporal synchronization, state dependent modulation, and multiple processing paths. The network combines the output of individual detection techniques in a physiologically motivated fashion which achieves improved object detection. The information fusion and object detection

properties of the image fusion network were demonstrated on mammograms and forward looking infrared (FLIR) images.

The PCNN fusion network provides a method of improving object detection accuracy by fusing the outputs of multiple object detection algorithms. For the example images, the accuracy of the fusion network surpassed the accuracy provided by the results of any single filtered output, or the logical AND of all filter results. The network takes pixel-based information as an input and produces an object-based output. The brightness values of the objects in the output image represent the degree to which each object matches the characteristics of the desired object. The PCNN fusion network provides a good foundation for implementing and evaluating other biological vision processing principles as more is learned about the primate vision system.

The calibration phase of this system is time consuming due to the complexity of setting the many PCNN parameter values. Once calibrated, the system requires no further attention and can be run autonomously. This large time requirement for parameter setting is typical of PCNN-based systems. The following chapter provides a remedy to this problem by developing the first adaptive PCNN. Given only an input and a desired output, the adaptive PCNN finds the parameter values necessary to approximate the desired output. This adaptive PCNN saves time and produces near-optimal settings for each parameter.

V. *Adapting PCNN Parameters*

5.1 Overview

A PCNN with a linking radius of 1 and a feeding radius of 1 contains 25 adjustable parameters (8 constants and 17 weights). A PCNN with a linking and feeding radius of 10 contains 889 adjustable parameters. Manually adjusting one or two parameters is feasible, but finding a near-optimal setting for all parameters requires searching a high dimensional parameter space. A task of this magnitude is best performed in an automated fashion. No PCNN presently exists which can adapt its parameters to meet a desired goal. Little guidance exists for selecting PCNN parameter values, and no guidance exists for adjusting poor parameter values to make the PCNN better achieve a goal (23, 46, 45, 47).

Training rules or parameter setting equations, exist for the multi-layer perceptron, Hopfield network, and many other neural networks (77). Some of the attributes that make the PCNN unique are the same attributes that have hindered development of equivalent equations or training rules for the PCNN. Linking connections within the PCNN causes the output of each neuron to be dependent upon the outputs of neighboring neurons. Adjusting one neuron's parameters has a nonlinear effect on all neighboring neurons. Another hindrance to the adaptation task is the pulse-based nature of the PCNN. Pulses are used to transport information between PCNN neurons. Well-known adaptation methods such as error back-propagation and reinforcement are typically applied to networks that use persistent signals to transfer information. A continuous signal equivalent to the PCNN must be developed before such techniques can be applied. The phenomenon of pulse capture (pulse synchronization) adds additional complexity to the adaptation task. Pulse capturing is a nonlinear operation which needs to be considered when attempting to apply linear adaptation techniques used in other neural networks to PCNNs.

In this chapter, adaptation equations are developed for all parameters of the PCNN. These equations take into account the inter-neural dependencies, pulse-based information transfer, and

pulse coupling that occur within the PCNN. First, a simplified, mathematically equivalent, persistent signal PCNN neuron model is developed. From this model, a system equation is formulated that provides the input-to-output relations needed to apply a gradient descent-based adaptation method. Backward error propagation is applied to the PCNN system equation to derive parameter adaptation equations for each parameter. Some of the resulting equations are time varying and require adaptation after every time step during a discrete time simulation. This time dependency limits the usefulness of the equations and increases computational requirements. Additional knowledge of pulse capturing is applied to these equations to reduce them to a form which is not a function of time. This allows all adaptation equations to be applied after PCNN execution is complete. The post-execution nature of the equations allows adaptation to be added to an existing PCNN without any internal modifications. This is a definite advantage for those who wish to add adaptation to an existing PCNN implemented in hardware. Given only an input and a desired output, an adaptive PCNN can find near-optimal parameter values that will minimize the squared error between the actual output and the desired output.

5.2 A Simplified, Mathematically Equivalent PCNN Neuron

The PCNN uses the Eckhorn artificial spiking neuron which consists of three major units: the feeding input branch, the linking input branch, and the pulse generator. Examining the neuron in a function oriented format simplifies both the neuron equations and the diagrams. The functional units of the neuron are shown in Figure 27 in a visually simplified model. Since the feeding inputs are connected to a source with constant value, no feeding leaky integrators are needed. Each linking input is replaced by the time signal $L'_{ik}(t)$, which is the output of the corresponding linking leaky integrator in the full model. The simplified model makes the following two assumptions:

1. Each neuron fires only once then remains dormant until the PCNN is restarted
2. $\theta_0 = 0$.

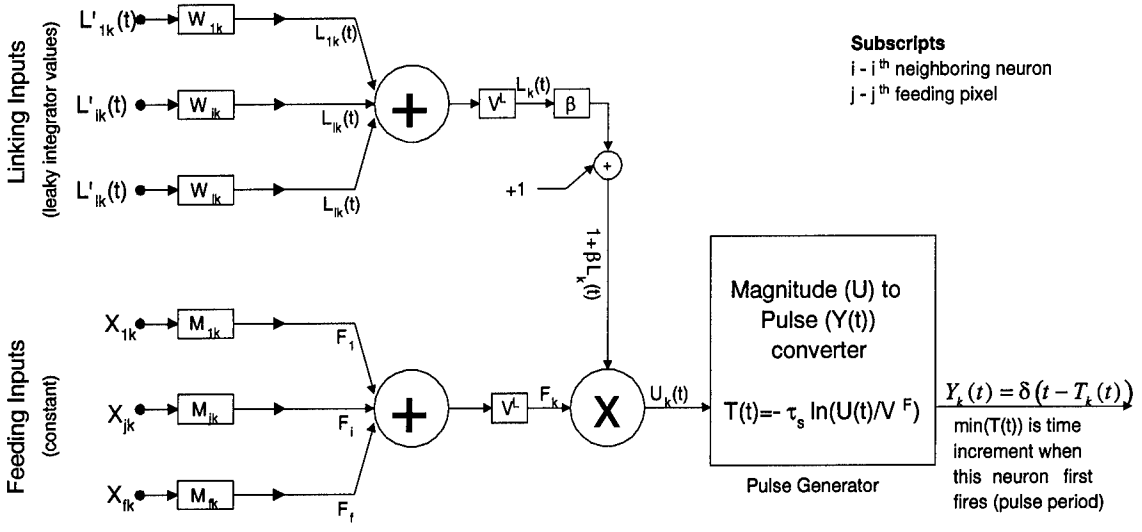


Figure 27 Simplified view of Eckhorn spiking neuron

The simplified neuron model is valid over the range $0 < U_k \leq V^S$. For $U_k > V^S$, $Y(t) = 1$ and for $U_k \leq 0$, $Y(t) = 0$ where $\delta(\cdot)$ is a unit impulse function. Within the bounds of these assumptions, the simplified model is mathematically equivalent to the full neuron model. The equations for the simplified model are:

Output Pulse:	$Y_k(t; X, M, L', W, \beta, V^F, V^L, V^S, \tau^S) = \delta(t - T_k(U_k))$
Output Pulse Period:	$T_k(t; X, M, L', W, \beta, V^F, V^L, V^S, \tau^S) = -\tau^S \ln\left(\frac{U_k}{V^S}\right)$
Total Input to Pulse Generator:	$U_k(t; X, M, L', W, \beta, V^F, V^L) = F_k(X)[1 + \beta L_k(t)]$
Total Feeding Input:	$F_k(X; M, V^F) = V^F \sum_{j=1}^{f} X_{jk} M_{jk}$
Total Linking Input:	$L_k(t; L', W, V^L) = V^L \sum_{i=1}^{l} L'_{ik}(t) W_{ik}$

where $T_k(t)$ defines the period of the output pulse produced by the k^{th} neuron. Combining the output pulse period (T_k) and total input (U_k) equations gives

$$T_k(t) = -\tau^S \ln\left(\frac{F_k(X)[1 + \beta L_k(t)]}{V^S}\right).$$

This is a good equation for providing a high-level understanding of the operation of the PCNN. It shows the output pulse period is a logarithmic function of the feeding inputs multiplied (modulated)

by the linking inputs. Adding in the total feeding input (F_k) and total linking input (F_k) equations produces

$$T_k(t) = -\tau^S \ln\left(\frac{V^F \sum_{j=1}^{f} X_{jk} M_{jk} (1 + \beta V^L \sum_{i=1}^{l} L'_{ik}(t) W_{ik})}{V^S}\right) \tag{12}$$

which is the equation for the period of the output pulse produced by the simplified Eckhorn neuron model. Additional expansion of Equation (12) could be performed by substituting

$$L'_{ik}(t) = \exp\left(\frac{-(t - T_i(t))}{\tau^L}\right) u(t - T_i(t)) \tag{13}$$

for the leaky integrator's output, or

$$L'_{ik}(t) = \delta(t - T_i(t)) \tag{14}$$

if no leaky integrator is used. $T_i(t)$ is the output pulse period of the i^{th} neuron connected to the linking inputs, $u(\cdot)$ is the unit step function, and $\delta(\cdot)$ is a unit impulse function. The value of $T_i(t)$ can be calculated using Equation 12 which creates a group of simultaneous equations, or can be replaced with the actual output value during PCNN execution.

5.3 Adapting Parameters Using Gradient Descent

Backward error propagation (backprop) is one of the most common techniques for developing adaptation rules for multilayer perceptron artificial neural networks (93, 80). Backward error propagation using gradient descent can be applied to Equation (12) to derive an adaptation equation for any chosen parameter. A gradient descent-based optimization technique requires an error functional to minimize. Defining this error functional as

$$E \equiv \frac{1}{2n} \sum_{k=1}^{n} (Desired_k - Actual_k)^2 \tag{15}$$

gives the mean squared error (MSE) between the desired output and the actual output of an n neuron PCNN. $Desired_k$ and $Actual_k$ are the desired and actual output of neuron i, respectively. This error functional actually defines MSE multiplied by 1/2. This scaling factor is included for convenience because it cancels other scale factors during equation derivation. It has no effect on the final adaptation results because minimizing MSE/2 also minimizes MSE. The partial derivative of this error functional with respect to a chosen variables provides the gradient of the output error with respect to that variable. Adjusting the chosen variable in the direction of the steepest descent of this error gradient will reduce the output error. This method of adaptation is known as the first-order gradient steepest descent method which is commonly used in artificial neural network weight update rules (77). The general PCNN adaptation rule for a variable is

$$G_i^{new} = G_i^{old} - \eta' \frac{\partial E}{\partial G_i}(G^{old})$$

where G^{old} is the variable before adaption, G^{new} is the variable after adaption, i is the index of the element of G that is being adapted, and η' is the adaptation rate (a small real valued number). This equation defines an adaptation rule for minimizing the squared error over n neurons. The partial derivative term provides the direction of the steepest descent with respect to the scalar G. Focusing on the k^{th} neuron within the PCNN, the error functional (Equation 15) becomes

$$E = \frac{1}{2}(Desired_k - Actual_k)^2. \tag{16}$$

Taking the partial derivative of this error functional for a single neuron gives

$$\frac{\partial E}{\partial G_i}(G^{old}) = -(Desired_k - Actual_k)\frac{\partial Actual_k}{\partial G_i}(G^{old})$$

where k is the index of the neuron.

The output pulse period of the simplified Eckhorn neuron (Equation (12)) can be substituted for the variable $Actual_k$ so that additional decomposition can be performed. There is no need to perform additional decomposition on the first occurrence of the variable $Actual_k$, thus it is left unchanged. The second occurrence of $Actual_k$ is substituted because the gradient term of the equation is decomposed to derive adaptation equations. Performing this substitution produces the equation.

$$\frac{\partial E}{\partial G_i}(G^{old}) = -(Desired_k - Actual_k)\frac{\partial T_k}{\partial G_i}(G^{old}) \tag{17}$$

A chosen variable within the k^{th} PCNN neuron can be adapted to reduce output error using the equation

$$G_i^{new} = G_i^{old} + \eta(Desired_k - Actual_k)\frac{\partial T_k}{\partial G_i}(G^{old}) \tag{18}$$

where $\eta = \frac{\eta'}{n}$ (to minimize the number of scaling variables that appear in future equations).

5.4 Applying Gradient Descent to PCNN Parameters

5.4.1 Feeding Weights Adaptation. The adaptation equation for the feeding weights is derived first because of its simple and straight forward nature. The previous section developed generalized adaptation equations based on gradient descent. These equations must be solved for a specific variable before they can be applied. Substituting M_{jk} for the general parameter G_i in Equation (17) results in the error gradient equation for the jk^{th} feeding weight

$$\frac{\partial E}{\partial M_{jk}} = -(Desired_k - Actual_k)\frac{\partial T_k}{\partial M_{jk}}. \tag{19}$$

Expanding the partial derivative of T_k with respect to M_{jk} by replacing T_k with Equation (12) gives

$$\frac{\partial T_k(t)}{\partial M_{jk}} = -\frac{\partial}{\partial M_{jk}}\tau^S \ln\left(\frac{V^F(\sum_{j=1}^{f} X_{jk}M_{jk})(1+\beta V^L \sum_{i=1}^{l} L'_{ik}(t)W_{ik})}{V^S}\right)$$

$$= \frac{-\tau^S X_{jk}}{\sum_{i=1}^{f} X_{ik} M_{ik}}.$$

Substituting this partial derivative term back into the error gradient equation for the jk^{th} feeding weight (Equation 19) gives the gradient of the output error with respect to the feeding weight M_{jk}

$$\frac{\partial E}{\partial M_{jk}} = (Desired_k - Actual_k) \frac{\tau^S X_{jk}}{\sum_{i=1}^{f} X_{ik} M_{ik}}.$$

Substituting this parameter specific information into the general adaptation Equation (18) produces

$$M_{jk}^{new} = M_{jk}^{old} - \eta (Desired_k - Actual_k) \frac{\tau^S X_{jk}}{\sum_{i=1}^{f} X_{ik} M_{ik}^{old}} \tag{20}$$

which is an adaptation equation for the j^{th} feeding weight (M_{jk}) of the k^{th} PCNN neuron to reduce output error.

5.4.2 Linking Weights Adaptation. The same procedure used to derive the feeding weight adaptation equation can be applied to the linking weights. The error gradient equation for the i^{th} linking weight W_{ik} is

$$\frac{\partial E}{\partial W_{ik}} = -(Desired_k - Actual_k) \frac{\partial T_k(t)}{\partial W_{ik}}.$$

Expanding the partial derivative of $T_k(t)$ with respect to W_{ik} gives

$$\frac{\partial T_k(t)}{\partial W_{ik}} = \frac{-\tau^S \beta V^L L'_{ik}(t)}{1 + \beta L_k(t)}.$$

Substituting linking weight specific information into Equation (18) produces

$$W_{ik}^{new} = W_{ik}^{old} - \eta^L (Desired_k - Actual_k) \frac{\tau^S \beta V^L L'_{ik}(t)}{1 + \beta L_k(t)}$$

where η^L is the adaptation rate for the linking weights. This adaptation equation is usable, but contains the time varying signals $L'_{ik}(t)$ and $L_k(t)$. Any adaptation performed using this equation must be performed at each timestep, or the value of $L'_{ik}(t)$ and $L_k(t)$ for each timestep must be saved for later processing.

Even though the adaptation equation for W_{ik} is a function of t, adaptation need not be performed for every value of t. The time varying aspect of the equation can be removed using the initial assumptions and knowledge of pulse capturing. The goal of this discussion is to replace the variable t with a constant that may differ for each neuron. The initial assumptions state a neuron can only pulse once. As before, let $Actual_k$ be the the actual output pulse period of the k^{th} neuron. Once execution is complete, the three following training possibilities exist:

if $Actual_k = Desired_k$ do not adapt

if $Actual_k < Desired_k$ adapt parameter to make neuron fire later

if $Actual_k > Desired_k$ adapt parameter to make neuron fire earlier.

Since all neurons pulse at $t = 0$, the neuron's firing time and output pulse period ($Actual_k$) are the same value and will be used interchangeably. Adapting the linking weights adjusts the degree to which the pulse period $Actual_k$ of the k^{th} neuron is influenced by the output of neighboring neurons. Pulse capture, which is the mechanism of neuron synchronization, only occurs in one direction (46). A neuron can only be captured by a neuron that fires earlier than itself because linking signals only exist from neurons that have fired. To make a neuron fire later one must lower the influence exerted by neighboring neurons that fire at $Actual_k$. To make a neuron fire earlier, the influence exerted by neighboring neurons that fire earlier than $Actual_k$ must be increased. Since the goal of this discussion is to replace all occurrences of t within the equation with a neuron specific time, let $\hat{t}_k$ represent that time for the k^{th} neuron. The time $\hat{t}_k$ is the time at which training must take place. This is the time at which the influential signals are present. This time is always the earlier of $Actual_k$ and $Desired_k$ (i.e., $\hat{t}_k = \min\{Desired_k, Actual_k\}$ where min is the minimum

operator). Substituting this earlier time $\hat{t}_k$ for t gives

$$W_{ik}^{new} = W_{ik}^{old} - \eta^L (Desired_k - Actual_k) \frac{\tau^S \beta V^L L'_{ik}(\hat{t}_k)}{1 + \beta L_k(\hat{t}_k)}. \tag{21}$$

$L'_{ik}(\hat{t}_k)$ and $L_k(\hat{t}_k)$ could each be written as a function of the linking inputs by performing a variable substitution using Equation (13). The variable $T_i(t)$ in Equation (13) is time varying and requires a time independent replacement. After execution is complete, the output pulse periods for all neurons are known. The output pulse period equation $T_i(t)$ for neuron i can be replaced by its actual output pulse period $Actual_i$. Making this substitution produces the leaky integrator output equation

$$L'_{ik}(\hat{t}_k) = \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i). \tag{22}$$

Expanding Equation (21) using Equation (22) produces

$$W_{ik}^{new} = W_{ik}^{old} - \eta^L (Desired_k - Actual_k) \frac{\tau^S \beta V^L \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i)}{1 + \beta V^L \sum_{j=1}^{l} \exp\left(\frac{-(\hat{t}_k - Actual_j)}{\tau^L}\right) u(\hat{t}_k - Actual_j)}. \tag{23}$$

If leaky integrators are not used on the linking inputs, the equation reduces to

$$W_{ik}^{new} = W_{ik}^{old} - \eta^L (Desired_k - Actual_k) \frac{\tau^S \beta V^L \delta(\hat{t}_k - Actual_i)}{1 + \beta V^L \sum_{j=1}^{l} \delta(\hat{t}_k - Actual_j)}. \tag{24}$$

Equations (23) and (24) are adaptation equations for modifying linking weight W_{ik} of the k^{th} PCNN neuron to reduce total output error. Either equation can be applied after all timesteps are completed knowing only the desired and actual output of the PCNN.

Not all desired outputs can be achieved by adapting the linking weights. A neuron cannot be influenced to fire at an earlier timestep through linking weight adaptation if no neighboring neuron fires at that timestep or earlier (if linking leaky integrators are used). Other parameters (feeding weights, etc.) must be adapted to achieve this goal. Another problem that may be

encountered when training linking weights is oscillations. Through linking, the neuron outputs become interdependent (6). A change in output of one neuron as it adapts based on a neighbor's output may cause that neighbor's output to change. During testing, this condition has caused oscillations between neighboring neurons that are adapting to achieve mutually exclusive goals. This problem can be controlled by decreasing η^L over successive training epochs. The neuron adaptation goals will remain mutually exclusive, but the cause of the oscillations is now damped and will cause the oscillations to diminish over successive training epochs.

Figures 28 and 29 present examples in which the linking weights are adapted to cause the center neuron to fire at earlier and later timesteps, respectively. Figure 28 shows the output of a PCNN before and after adapting the linking weights. Darkened circles represent pulsing neurons and empty circles represent non-pulsing neurons. Output timestep and output pulse period are synonymous (i.e., a neuron that pulses at time $t = 3$ has an output pulse period of $T = 3$). The PCNN is composed of a 3×3 array of neurons connected using a linking radius of 1. Each neuron has a single feeding input. An identical set of linking weights is used for each neuron and all linking weights are initially set to 1. With these linking weights all neurons with non-zero feeding inputs fire together at $t = 2$. The desired output is the center neuron firing at $t = 3$ and all other neurons, with non-zero feeding inputs, firing at $t = 2$. Adaptation was performed by averaging the needed weight changes and applying the average to all neurons. The value of η^L is initially set to 0.1 and decreased over time. After several adaptation runs, the desired output is achieved. The weights connecting the center neurons to the capturing neurons have decreased to the point where the center neuron is no longer captured. The network adapted to cause the center neuron to fire at a later timestep. Figure 29 shows another training run using the same network. The goal in this run is to adapt the PCNN to cause the center neuron to fire at an earlier timestep than it originally fires. The initial parameters used in this run are the same as in the previous run. After adaptation, the center neuron fires at the desired timestep. The linking weights connected to pulsing neighboring neurons have increased to a value which allows the center neuron to be captured by these neurons.

A point worth noting: for this input, the PCNN linking weights cannot be adapted to cause the center neuron to pulse at timestep 3 without the use of leaky integrators. Linking works through the mechanism of pulse capture (46), and no neurons pulse at timestep 3 which could influence the center neuron through pulse capture.

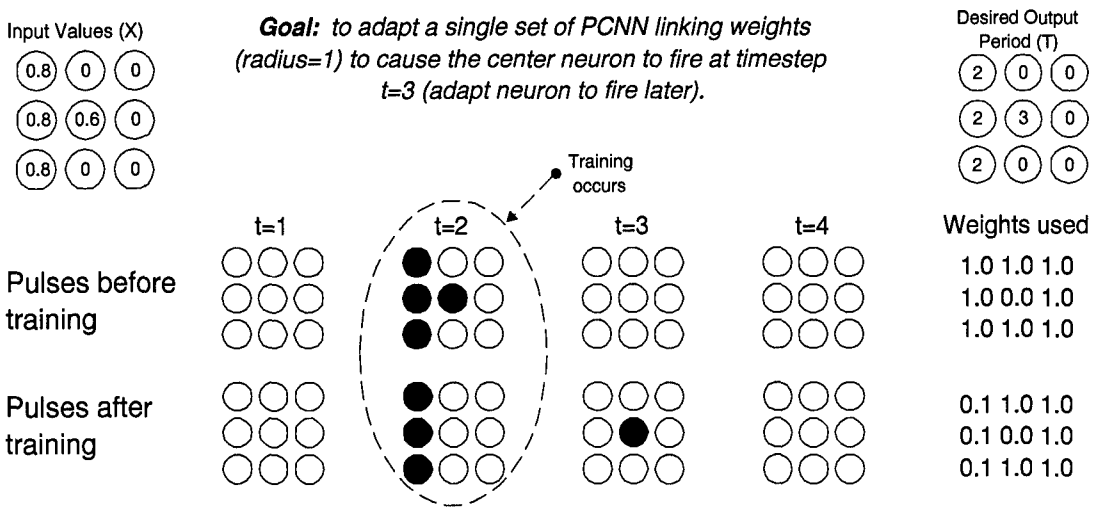


Figure 28 PCNN adaptation example: Linking weights are adapted to cause center neuron to fire at later timestep

5.4.3 Global Linking Strength (β) Adaptation. Of the many PCNN variables, β is most likely to be adjusted since it directly effects the coarseness of any segmentation that is performed. Using the same procedure used to derive the previous parameter adaptation equations, the partial derivative of the output equation with respect to β is

$$\frac{\partial T_k}{\partial \beta} = \frac{-\tau^S L_k(t)}{1 + \beta L_k(t)}.$$

The adaptation equation for β is

$$\beta^{new} = \beta^{old} - \eta^{\beta}(Desired_k - Actual_k)\frac{\tau^S L_k(t)}{1 + \beta L_k(t)}$$

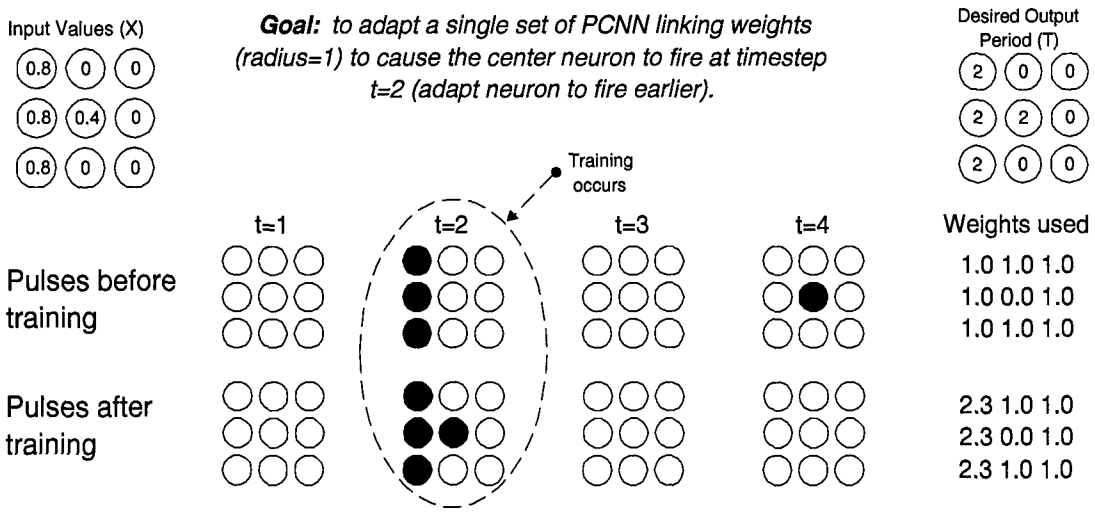


Figure 29 PCNN adaptation example: Linking weights are adapted to cause center neuron to fire at earlier timestep

where η^{β} is the adaptation rate for β. This equation is concise and usable, but contains the internal time varying signal $L_k(t)$ that must be processed at time t or stored for later use. Performing variable substitution using Equation (22) gives

$$\beta^{new} = \beta^{old} - \eta^{\beta}(Desired_k - Actual_k)\frac{\tau^S V^L \sum_{i=1}^{l} \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i)}{1 + \beta V^L \sum_{i=1}^{l} \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i)}. \quad (25)$$

If leaky integrators are not used on the linking inputs, the equation reduces to

$$\beta^{new} = \beta^{old} - \eta^{\beta}(Desired_k - Actual_k)\frac{\tau^S V^L \sum_{i=1}^{l} \delta(\hat{t}_k - Actual_i)}{1 + \beta V^L \sum_{i=1}^{l} \delta(\hat{t}_k - Actual_i)}. \quad (26)$$

Equations (25) and (26) are adaptation equations for modifying the β of the k^{th} PCNN neuron to reduce output error.

The parameter β performs a function very similar to the linking weights. The value of β could be incorporated into the linking weights, but is usually kept separate for the convenience of having a single variable that controls linking strength. When adapting the linking weights, there is no need to adapt β because linking strength is inherently included in the weight magnitudes. As

with adapting linking weights, not all desired outputs can be achieved by adapting β alone. The possibility of training oscillations exist when adapting β for neurons that have mutually exclusive goals.

Figure 30 shows the results of adapting β. This example uses the same network architecture used in the linking weight adaptation example. The goal is to adapt β to cause the center neuron to fire at timestep $t = 2$. Using Equation (13) and solving the simultaneous equations shown in 7 for the nine neuron PCNN shows the desired output can only be achieved using a value of β in the interval $[0.03659, 0.05807]$. For the first training case, the network is started with $\beta = 0.1$ which is too large to achieve the goal. The network adapts until it reaches $\beta = 0.05749$ which induces the center neuron to fire at $t = 2$. The second training case covers the opposite situation where the network is started with $\beta = 0.001$ which is too small. The network increases β until it reaches $\beta = 0.03840$ which satisfies the goal. In both cases $\eta^{\beta} = 0.0001$ is used as the learning rate.

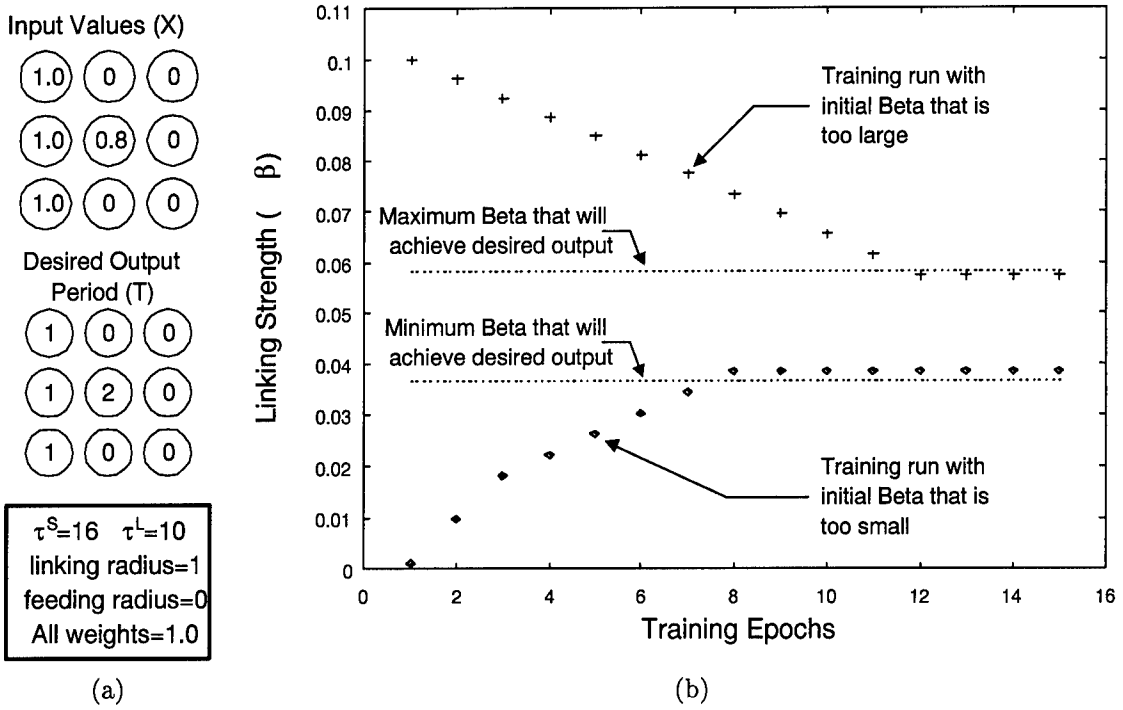


Figure 30 PCNN adaptation example: Beta (β) is adapted to cause center neuron to fire at timestep t=2. (a) Input and desired output for PCNN. (b) Adaptation of β during two training runs. Upper plot starts with β that is too large, lower plot starts with β too small.

5.4.4 Pulse Generator Time Constant (τ^S) Adaptation. The pulse generator time constant τ^S is another PCNN constant that is likely to be adjusted to achieve a desired output. It determines the timestep to which an input value will be mapped if no linking influence exist. If the value τ^S is smaller than is required to ensure a one-to-one mapping from input values to timesteps, τ^S will directly effect the coarseness of any segmentation that is performed. Using the same procedure used to derive the previous parameter adaptation equations, the adaptation equation for τ^S is

$$\begin{aligned} \tau^S_{new} &= \tau^S_{old} - \eta^{\tau^S}(Desired_k - Actual_k) \times \\ &\quad \ln\left(\frac{V^F \sum_{j=1}^{f} X_{jk} M_{jk}(1 + \beta V^L \sum_{i=1}^{l} \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i) W_{ik})}{V^S}\right) \end{aligned} \tag{27}$$

where η^{τ^S} is the adaptation rate for τ^S. If leaky integrators are not used on the linking inputs, the equation reduces to

$$\begin{aligned} \tau^S_{new} &= \tau^S_{old} - \eta^{\tau^S}(Desired_k - Actual_k) \times \\ &\quad \ln\left(\frac{V^F \sum_{j=1}^{f} X_{jk} M_{jk}(1 + \beta V^L \sum_{i=1}^{l} \delta(\hat{t}_k - Actual_i) W_{ik})}{V^S}\right). \end{aligned} \tag{28}$$

5.4.5 Linking Leaky Integrator Time Constant (τ^L) Adaptation. The linking leaky integrator time constant is an interesting parameter, because at first glance a leaky integrator would appear to serve no function in a pulse-once scenario. If only one pulse is emitted by each neuron, then there is no need for a leaky integrator to accumulate pulses. The leaky integrator does serve another function. It converts the single pulse to a persistent (albeit decaying) signal. This persistent signal allows a neuron to be influenced by an earlier firing neighboring neuron even if it is not captured by that neuron. What does this mean to the overall PCNN operation? It allows neighboring neurons to influence a neuron to fire early, but at a timestep in which no neighboring neuron is firing. This is an important fact. Without linking leaky integrators, the center neuron

in the second linking weight training example (Figure 29) can fire at only timestep 2 or timestep 4. It either *is* or *is not* captured by the three neighboring neurons. With leaky integrators, the influence of the three neighboring neurons is still present at timestep 3, thus the center neuron can be influenced to fire at timestep 3. With that said, the same procedure used to derive the previous parameter adaptation equations can be used to derive an equation for adapting τ^L. The adaptation equation for τ^L is

$$\tau^L_{new} = \tau^L_{old} - \eta^{\tau^L}(Desired_k - Actual_k) \times \frac{\tau^S \beta V^L \sum_{i=1}^{l} W_{ik} \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right)\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i)}{1 + \beta V^L \sum_{i=1}^{l} \exp\left(\frac{-(\hat{t}_k - Actual_i)}{\tau^L}\right) u(\hat{t}_k - Actual_i)}. \quad (29)$$

where η^{τ^L} is the adaptation rate for τ^L. This same procedure can be applied to the feeding leaky integrators, but will not be since the simplified Eckhorn neuron contains none.

5.4.6 Feeding (r^F) and Linking Radius (r^L) Adaptation. The feeding and linking radius determine the number of input values or neighboring neuron outputs that are processed by a single neuron. No adaptation equations will be derived for these parameters because they are inherently adapted by the feeding weight and linking weight adaptation equations. For a given feeding and linking radius, the corresponding weight adaptation equation will adjust the radius by adjusting weights of undesirable inputs towards zero. This will effectively reduce the radius if a reduced radius is needed to achieve the desired output. The weight adaptation equations cannot increase the radius if a larger radius is required. During training, a sufficiently large radius should be selected to achieve the desired output. Equations that can be used to determine a sufficient radius are given in Ranganath (74).

5.4.7 Using Identical Parameter Values for Multiple Neurons. The adaptation equations derived above can be directly applied to individual neurons to achieve a near-optimal setting for each neuron. Individual parameter settings for each neuron will train a network to operate well on

a single image or a group of images with spatially similar content. When a network with spatial invariance is desired, identical parameter settings are often used for all neurons in the network (or network layer). The adaptation equations presented in this dissertation can be applied to this type of network by summing the needed update values and dividing by the total number of neurons in the PCNN. This summed and scaled update value can then be applied to each neuron in the PCNN. Averaging the needed update values may cause oscillations because a single neuron can cause the same magnitude change to a parameter's value as multiple neurons.

5.4.8 Limitations of the Gradient Descent Method. As with all search techniques based solely on gradient descent, the adaptation equations presented in this research may find a local minima in the error surface and not reach a global minimum MSE. The quality of the results is directly dependent upon the shape of the error surface and the initial parameters. Often multiple training runs using randomly chosen initial values are used to reduce the effect of local minima. Genetic algorithms and simulated annealing have also been successfully used to reduce the effect of local minimas while maintaining most of the efficiency associated with gradient base searches (31, 81, 87, 70). Only a true global search or absolute knowledge of the error surface can guarantee an optimal result.

5.5 Setting the Remaining Parameters

The remaining parameters, which include the firing threshold offset (θ_0), magnitude adjustment constants (V^F, V^L, and V^S), and maximum timestep, can be viewed as neuron tuning parameters. They are no less important than the other parameters, but serve a slightly different purpose. These remaining parameters control internal signal levels which alter the efficiency and resolution of the PCNN processing. Sub-optimal values for these parameters will result in inefficient processing or distorted output.

5.5.1 The Pulse Generator Firing Threshold (θ_0). To set the pulse generator firing threshold to prevent any values of U_k less than or equal to 0.6 from generating a pulse, set $\theta_0 = 0.6$. As implemented, the threshold has the performance side effect described earlier. For $\theta_0 \neq 0$, all adaptation equations will work properly, but equations for setting V^S will need modification to include θ_0. Any other constants used to compensate for the side effect of θ_0 will become interdependent with θ_0. We recommend setting $\theta_0 = 0$ and either threshold the input before PCNN execution, or threshold the output after PCNN execution.

5.5.2 The Magnitude Adjustment Constants (V^F, V^L, and V^S). As previously stated, the pulse generator operates over the input range [0,V^S]. The magnitude adjustment constants V^F and V^L are used to scale magnitudes of F_k and L_k, respectively, to produce a value of U_k that is within this desired range. V^S can also be set to scale the value of U_k. For optimal scaling, the variables should be set to any combination that satisfies

$$F_{max}(1 + \beta L_{max}) = V^S$$

where F_{max} and L_{max} are the maximum possible values of F_k and L_k, respectively. Expanding this equation to contain V^F and V^L gives

$$V^F(\sum_{j=1}^{f} X_{jk}M_{jk})_{max}(1 + \beta V^L(\sum_{i=1}^{l} \exp\left(\frac{\hat{t}_k - Actual_i}{\tau^L}\right) u(\hat{t}_k - Actual_i)W_{ik})_{max}) = V^S$$

where the subscript *max* denotes the maximum possible value. This equation simply states that all constants should be set to maintain $(U_k/V^S) \leq 1$.

Another use for V^L can be to maintain a consistent total linking strength when the linking radius is changed. For example, a neuron with a linking radius of 1 is connected to eight neighboring neurons and can receive up to eight simultaneous pulses. If the linking radius is changed to 3, the

same neuron can now receive up to 48 simultaneous pulses. V^L can be adjusted to make the total linking magnitude the same for each case. This allows the same β to be used for both cases.

5.5.3 The Maximum Timestep. This is not a true PCNN parameter, but is mentioned here to aid efficiency. The maximum timestep is defined as the total number of timesteps a PCNN should be executed during a training epoch. This number should be set equal to the largest timestep present in the desired output. Any lesser value prevents the adaptive PCNN from fully approximating the desired output. Any greater value will result in unnecessary processing since the extra time steps can never match anything in the desired image.

5.6 Parameter Adaptation Example Using an MRI

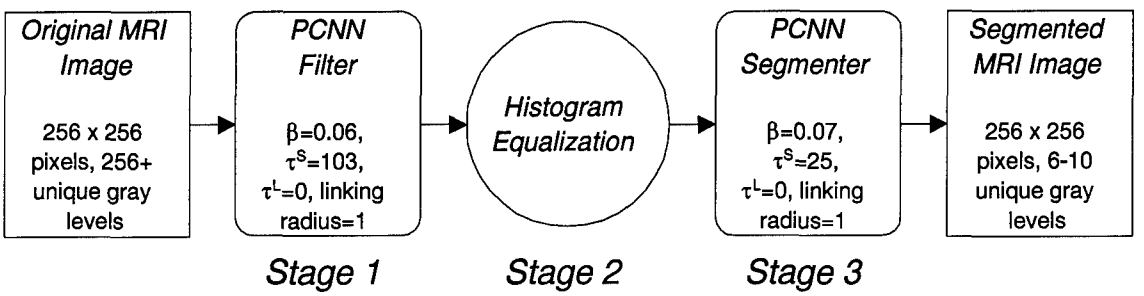
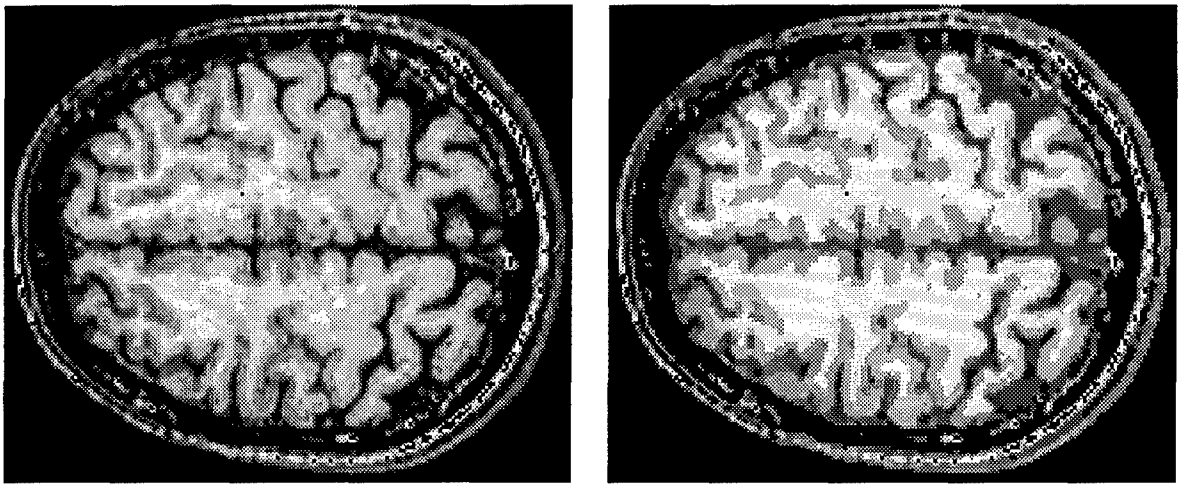


Figure 31 PCNN-based process used to segment MRIs for 3D modeling.

To demonstrate the utility of the adaptive PCNN, it is used to find the parameter settings necessary to segment MRIs with the PCNN. Another research effort at the Air Force Institute of Technology (AFIT) uses the process in Figure 31 to segment MRIs for the purpose of 3D modeling (2). For this process, all neurons within a PCNN use identical parameters. The PCNN filter and segmenter stages are described in detail in Ranganath (74) and Johnson (46). The original images contain 256+ unique gray levels and the segmentation process groups similar pixels to form an output image with only 6-10 unique gray levels.

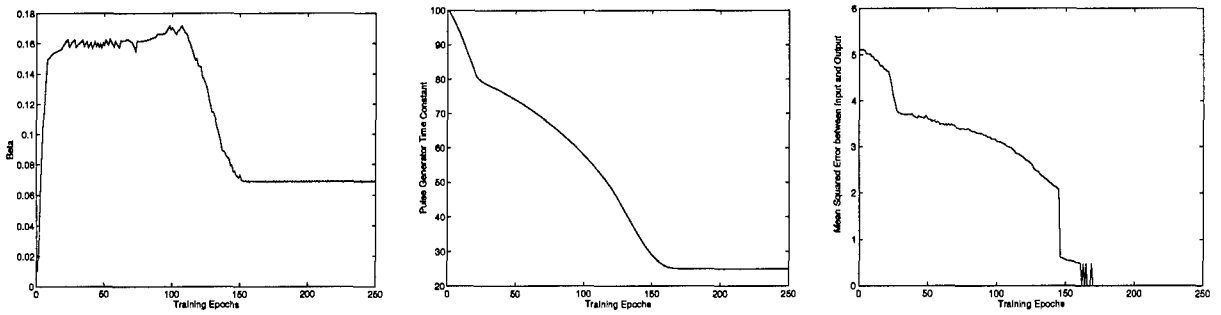


(a) (b)

Figure 32 Input and output of the PCNN in Stage 3 of the MRI segmentation process (256-by-256 pixel MRI, $\beta = 0.07$, and $\tau^S = 25$). (a) Input image containing 45 unique intensity levels (b) output image containing 7 unique intensity levels.

The first adaptation example demonstrates the adaptive PCNN is capable of finding the parameters necessary to produce an output that is within the PCNN's capabilities. The easiest way to demonstrate this point is to take the output of another PCNN and have the adaptive PCNN find the parameters needed to produce that output. Figure 32 shows the input and output of the PCNN in Stage 3 of the MRI segmentation process shown in Figure 31. All pixels in the input image are non-zero (even the dark background). The output image has been converted from timestep values to intensities for viewing purposes.

The adaptive PCNN is given the input to Stage 3, the desired output, and the arbitrary initial conditions of $\beta = 0.01$, and $\tau^S = 100$. The values $\beta = 0.07$ and $\tau^S = 25$ are used to create the desired output. The goal of this example is for the adaptive PCNN to minimize the squared error between its output and the desired output by adapting the parameters β and τ^S. Figure 33 shows the parameters as the PCNN adapts to minimize the squared error. The parameters were adapted to $\beta = 0.07$, and $\tau^S = 25$ and the final squared error was driven to zero. The desired output was reproduced with 100% accuracy. Several adaptation runs were performed using various



(a) (b) (c)

Figure 33 Adaptive PCNN parameters during adaptation while approximating the processing performed by the PCNN in Stage 2 of the MRI segmentation process. Goal of training is to match the parameters ($\beta = 0.07$ and $\tau^S = 25$) and the output of the Stage 2 PCNN. (a) Beta (b) pulse generator time constant τ^S (c) squared error between desired output and adaptive PCNN output.

initial values of β and τ^S. In all cases the adaptive PCNN found the correct parameters resulting in a squared error of zero.

The second example attempts to approximate the entire MRI segmentation process with a single PCNN segmenter. In this example the desired output cannot be achieved by the PCNN with 100% accuracy. The PCNN segmenter cannot fully reproduce the filter actions performed in Stage 1 by the PCNN filter, or the brightness adjustments performed by the histogram equalization in Stage 2. Prior to running the adaptive PCNN, a manual attempt was made to have a single PCNN approximate this process. Manually adjusting the PCNN parameters is a time consuming process. After three days without success the manual attempt was abandoned and the task was given to the adaptive PCNN. The adaptive PCNN is provided the input to the entire MRI segmentation process, the desired output, and the arbitrary initial conditions of $\beta = 0.74$, $\tau^S = 47$, and $\tau^L = 6.0$. Figures 34a and 34b show the input image and the desired output. All pixels in the input image are non-zero (even the dark background). The goal of this example is to minimize the squared error between the actual and desired output by adapting β, τ^S and τ^L. Figures 34c and 34d show the actual output and the resulting squared error between the actual and desired output. Perceptually, the difference between the actual and desired output are small.

Table 6 Error between adaptive PCNN output and desired output. PCNN parameters were adapted to minimize error on the reference image only. These parameters were then used on the remaining images to determine their generalization properties.

Image	Squared Error	Pixels that differ	Pixels that differ by more than 1 gray level
Reference	0,058	10.71%	0.30%
Image 1	0.055	9.75%	0.44%
Image 2	0.053	9.51%	0.34%
Image 3	0.061	10.20%	0.64%
Image 4	0.084	12.64%	1.36%
Image 5	0.099	13.91%	1.98%
Image 6	0.074	12.62%	0.75%
Mean	0.071	11.44%	0.92%
Std. Dev.	0.018	1.85%	0.63%

Figure 35 shows the adaptation of the three parameters and the resulting mean squared error between the actual and desired output. As expected, the squared error was not driven to zero, but was significantly reduced to 0.058. The actual and desired outputs differ in 10.7% of their pixels. However, only 0.3% of the pixels differ by more than one gray level. These results reflect the fact that the adaptation equations were derived from an error term based on squared error. The results would differ if a different error term were defined.

In the third example, the results of the adaptive PCNN are examined for generalization properties. Will the parameters that minimized the squared error in one image produce similar results in similar images? Seven MRIs were processed using the complete MRI segmentation process and using the adaptive PCNN segmenter. The output images of the MRI segmentation process are used as the desired outputs. The adaptive PCNN was trained on the first image and the resulting parameters are used to process the remaining six images. Table 6 shows the squared error and pixel error (percent of pixels that differ) between the adaptive PCNN output and the desired output. The standard deviation across all images was less than 1.9%, showing the parameters generalize well. This example shows adaptation can be performed using a single image from a set of images, and the remaining images in the set can be processed with consistent results.

5.7 Summary

The equations for implementing a PCNN with self-adjusting parameters have been presented. Given a desired output, these equations adapt the PCNN parameters to minimize the mean squared error of the actual output. These adaptation equations cover all PCNN constants and weights. Both simple and complex examples of parameter adaptation are provided to demonstrate the utility of adaptation. For a given image, the segmentation produced by a PCNN with unknown parameters was reproduced with 100% accuracy. The multi-stage MRI segmentation process, which performed image manipulation beyond the capabilities of a PCNN, was approximated with only 10.7% of the pixels differing from the desired output and less than 0.3% differing by more than one gray level.

These adaptation equations save time and simplify using the PCNN. A researcher need only know the desired output and the adaptive PCNN will produce the parameters that best reach that goal. As demonstrated in the MRI example, self-adjusting parameters allow the PCNN's utility as a segmenter to be easily exploited on real world images. To process a set of images, simply execute the adaptive PCNN on a single image from the set. The adaptive PCNN will find the parameter values that best produce that desired output. These parameters generalize well and can be used on the remaining images in the set with consistent results.

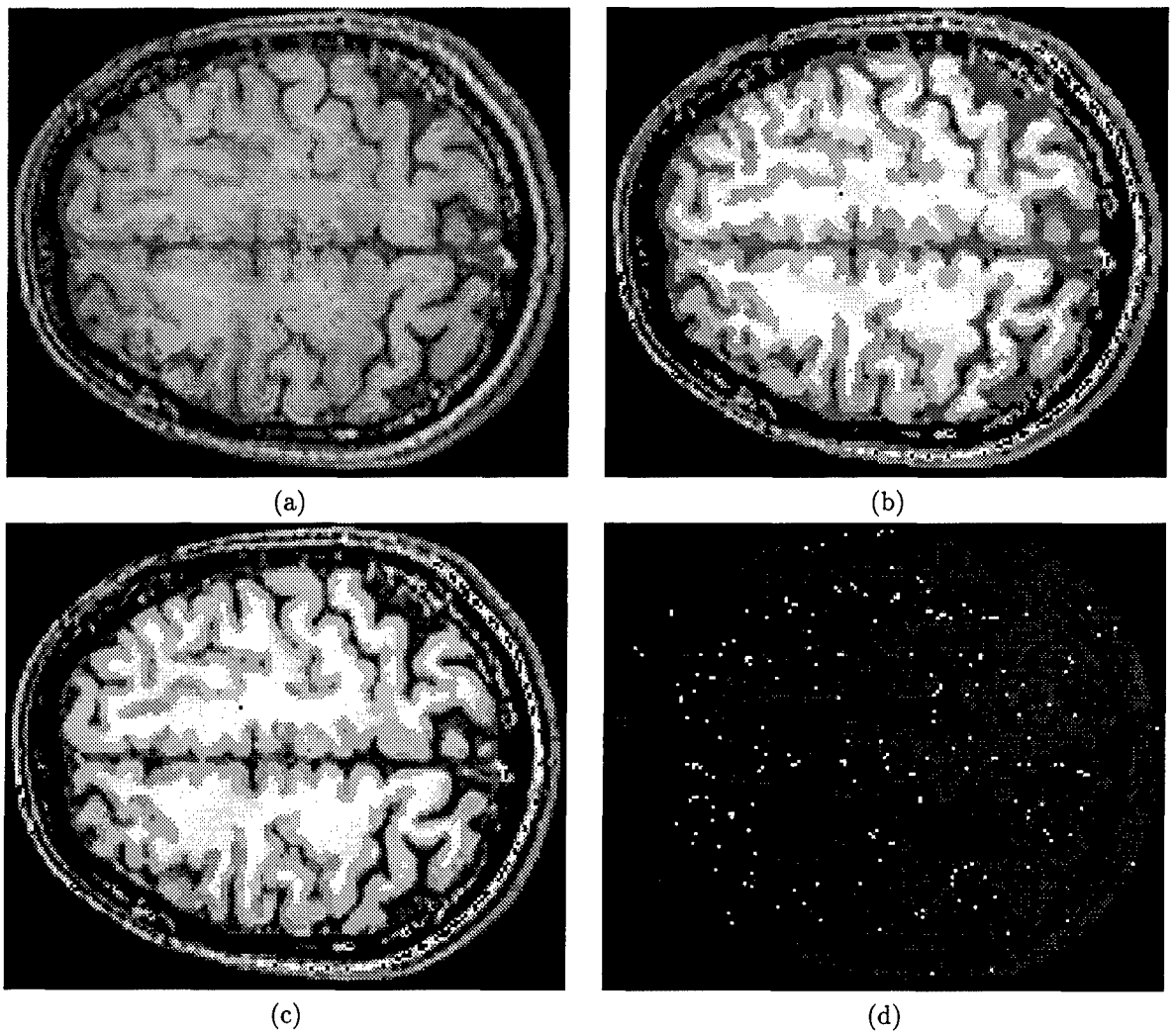


Figure 34 Adaptation example on a 256-by-256 pixel Magnetic Resonance Image (MRI). (a) Original image (b) desired output (c) output produced by adaptive PCNN after adaptation (d) the squared error at each pixel between desired output and adaptive PCNN output.

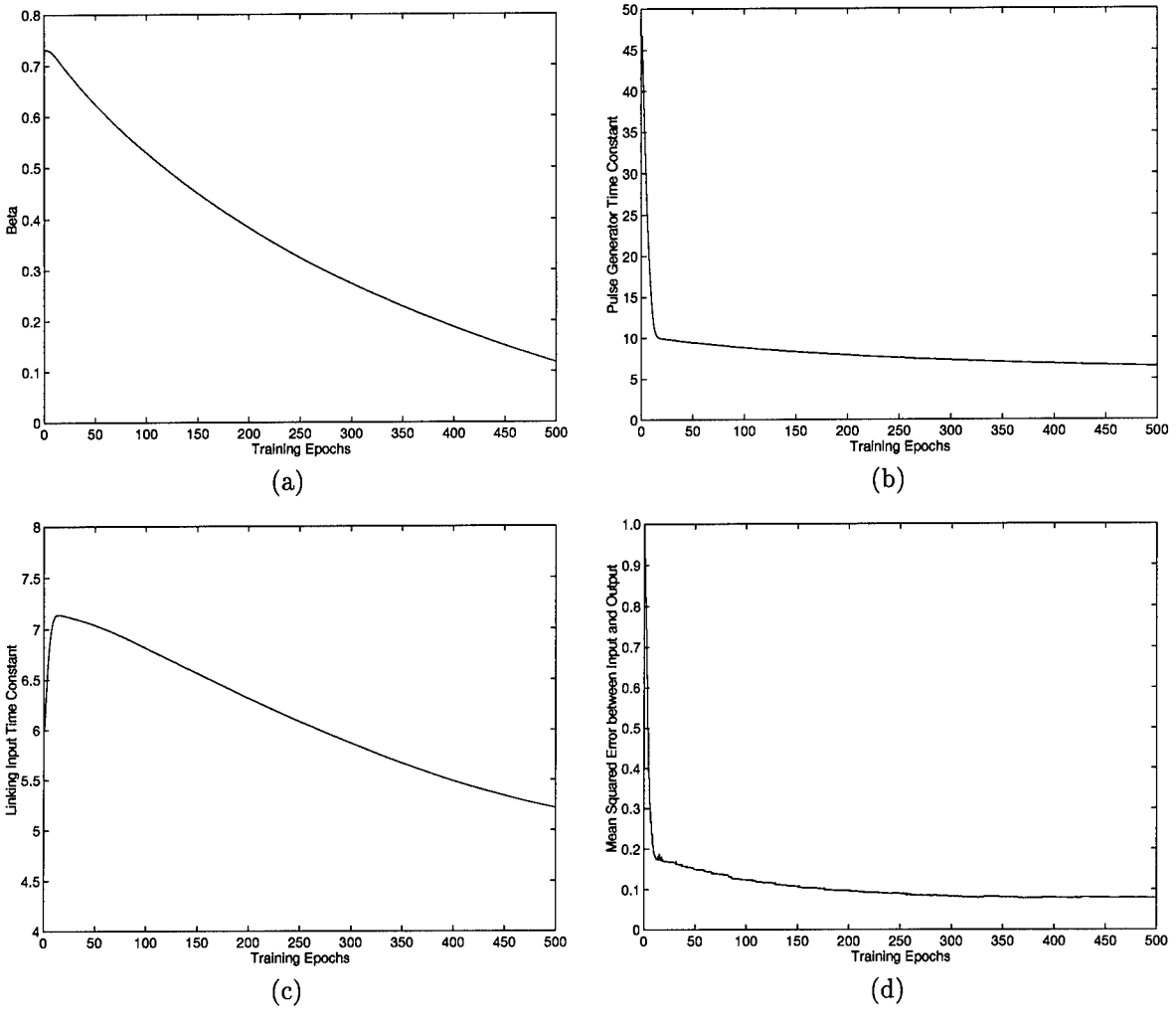


Figure 35 Adaptive PCNN parameters during adaptation on the 256-by-256 pixel MRI shown in Figure 34. (a) The global linking strength β (b) the pulse generator time constant τ^S (c) the linking input time constant τ^L (d) the mean squared error between the desired output and the adaptive PCNN output.

VI. Conclusion and Contributions

6.1 Conclusion

A new technique for modeling the primate vision system has been presented. For the first time, the theorized and biologically observed vision principles of spatial frequency filtering, multiple processing paths, competitive information processing, state dependent modulation, and temporal synchronization are brought together in a single model. Using these biologically-based principles, the PCNN feature extraction network performs spatial frequency analysis producing basic features for use in object detection and recognition. It can provide an effective, flexible, and extensible feature extraction stage for an object recognition system. Simple modifications have been presented that can extend the the model's capabilities to perform spatio-temporal (motion) and spatial wavelength (color) analysis. With these extended capabilities, the feature extraction model can simulate visual processing of many known basic information types (luminance, wavelength, direction, and orientation) processed by neuronal processing units in the early stages of the primate vision system. Cascading this model to simulate observed multi-layer hierarchical vision processing can produce the higher order moments of the basic information types such as texture and acceleration. This set of features provides a sufficient basis for nearly any type of visual object detection/recognition goal.

The PCNN image fusion network provides a novel and effective approach to information fusion. It provides a physiologically motivated method of associating dissimilar, spatially disjoint, features with objects. This model produces promising results in the areas of object detection and information fusion. The capabilities of the model have been demonstrated on real world images in the areas of breast cancer detection and automated target detection. The object detection accuracy of the network exceeds the accuracy of published detection systems.

The most significant contribution made by this research is the development of adaptation equations for the PCNN. These equations allow the near-optimal setting of PCNN parameters.

Researchers can now quickly and reliably use the PCNN as a research tool instead of spending time empirically setting the PCNN parameter values. Given only an input and a desired output, the adaptive PCNN will find all parameter values necessary to approximate that desired output. The adaptation equations automatically adapt parameter values to minimize squared error between the actual and desired output. To demonstrate its usefulness as a segmenter, the adaptive PCNN was used to segment actual magnetic resonance brain images.

6.2 Contributions

This research makes the following contributions:

1. *The first PCNN-based physiologically motivated feature extraction system.* This research applies primate vision processing principles such as spatial frequency filtering, state dependent modulation, temporal synchronization, competitive feature selection and multiple processing paths to create the first physiologically motivated, PCNN-based image fusion network. This is the first PCNN-based system to simulate feature extraction and attention focus observed in the biological vision system.

2. *The first PCNN-based physiologically motivated information fusion system.* This research develops the first PCNN-based information fusion network. Physiologically motivated information fusion theories are analyzed and implemented in this network. The network is used to fuse the results of several object detection techniques to improve object detection accuracy. The feature extraction and object detection properties of the image fusion network are demonstrated on mammograms and forward looking infrared (FLIR) images. The network removed 93 percent of the false detections without removing any true detections in the FLIR images and removed 46 percent of the false detections while removing only 7 percent of the true detections in the mammograms.

3. *The first adaptive PCNN.* Using gradient descent-based backward error propagation, this research develops the first fully adaptive PCNN. Given only an input and a desired output, the adaptive PCNN will find all parameter values necessary to best achieve that desired output. The adaptive PCNN automatically adapts parameter values to minimize squared error between the actual and desired output. To demonstrate its usefulness as a segmenter, the adaptive PCNN was used to segment MRIs of the brain. Adaptation was used to find parameter values that would cause the PCNN to approximate two Magnetic Resonance Image segmentation processes used in model-based vision research (2). For the given images, the adaptive PCNN reproduced the results of the first process with 100% accuracy and approximated the more difficult second process with 90% accuracy.

Bibliography

1. Abbot, L. F. "Decoding neuronal firing and modeling neural networks," *Quarterly Review of Biophysics*, *27*(3):291–331 (1994).
2. Abrahamson, Shane, et al. "Segmenting Magnetic Resonance Brain Images using Pulse Coupled Neural Networks," *Submitted to IEEE Transactions on Medical Imaging* (January 1997).
3. Albrecht, D. G. "Visual cortical neurons: are bars or gratings the optimal stimuli?," *Science*, *207*:88–90 (1980).
4. Allen, F. T., et al. "Scene evaluation using a pulse-coupled neural network (PCNN)," *SPIE Proceedings* (1995).
5. Allen, F. T., et al. "A neural bridge from syntactic to statistical pattern recognition," *Optical Memory & Neural Networks, vol.5, no.2, p. 79-89* (1996).
6. Broussard, Randy, et al. "Physiologically Motivated Image Fusion for Object Detection using a Pulse Coupled Neural Network," *Accepted by IEEE Transactions on Neural Networks* (1997).
7. Burns, Thomas J. "Stephen Grossberg's Boundary Contour System." PHD Minor Exam Requirement, 1992.
8. Capocaccia, G., et al. "Data fusion approach to obstacle detection and identification," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.1003, p. 409-19* (1989).
9. Carpenter, G. A. and S. Grossberg. "Dynamic models of neural systems: propagated signals, photoreceptor transduction, and circadian rhythms," *Oscillations in Mathematical Biology Proceedings, p. 196, 102-96* (1983).
10. Carpenter, G. A. and S. Grossberg. "A neural theory of circadian rhythms: the gated pacemaker," *Biological Cybernetics, vol.48, no.1, p. 35-59* (1983).
11. Carpenter, G. A., et al. "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, *3*(5):698–712 (1992).
12. Carpenter, G. A., et al. "Invariant recognition of cluttered scenes by a self-organizing ART architecture: Cort-X boundary segmentation," *Neural Networks*, *2*:169–181 (1989).
13. Cohen, M. A. and S. Grossberg. "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Transactions on Systems, Man and Cybernetics, vol.SMC- 13, no.5, p. 815-26* (1983).
14. Crick, F. and C. Kosch. "The problem of consciousness." *Mind and Brain* 125–136, W. H. Freeman and Company, 1993.
15. Daugman, John G. "Two dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, *20*:847–856 (1980).
16. Daugman, John G. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, *2*(7):1160–1169 (1985).
17. Dauk, Ronald, et al. "A Comparative Study of Texture Measures for the Classification of Breast Tissue," *Submitted to IEEE Transactions on Systems Man and Cybernetics* (1996).

18. der Smagt, P. P. Van. "Minimization methods for training feedforward neural networks," *Neural Networks*, *7*(1):1–11 (1994).

19. DeValois, K. K. "Spatial vision based upon color differences," *SPIE*, *2054*:95–103 (1994).

20. DeValois, K. K., et al. "Responses of striate cortex cells to grating and checkerboard patterns.," *J Physiol (London)*, *291*:483–505 (1979).

21. Eckhorn, R., et al. "Coherent oscillations: a mechanism of feature linking in the visual cortex? Multiple electrode and correlation analyses in the cat," *Biological Cybernetics, vol.60, no.2, p. 121-30* (1988).

22. Eckhorn, R., et al. "Feature linking via stimulus-evoked oscillations: experimental results from cat visual cortex and functional implications from a network model," *IJCNN: International Joint Conference on Neural Networks (Cat. No.89CH2765-6), p. 2 vol. (790+646), 723-30 vol.1* (1989).

23. Eckhorn, R., et al. "Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex," *Neural Computation*, *2*:293–307 (1990).

24. Eckhorn, R., et al. "Feature linking across cortical maps via synchronization," *Parallel Processing in Neural Systems and Computers, p. xv+626, 101-4* (1990).

25. Francis, Gregory and Stephen Grossberg. "Cortical Dynamics of Form and Motion Integration: Persistence, Apparent Motion, and Illusory Contours." Soon to be published in Vision Research, 1994.

26. Gabor, D. "Theory of communication," *J IEE (London)*, *93*:429–457 (1946).

27. Gaskill, Jack D. *Linear Systems, Fourier Transforms, and Optics*. New York: John Wiley & Sons, 1978.

28. Georgiopoulos, M., et al. "A survey of learning results in ART architectures," *unknown*, 1–9 (1995).

29. Giger, M. L. "Computer-Aided Diagnosis," *RSNA Categorical Course in Physics*, 283–298 (1993).

30. Glezer, V. D., et al. "Investigation of complex and hypercomplex receptive fields of visual cortext of the cat as spatial frequency filters.," *Vision Research*, *13*:1875–1904 (1973).

31. Golden, J., et al. "Evolutionary optimization of a neural network-based signal processor for photometric data from an automated DNA sequencer," *Evolutionary Programming IV. Proceedings of the Fourth Annual Conference on Evolutionary Programming, p. xx+805* (1995).

32. Gray, Charles M., et al. "Oscillatory Responses in Cat Visual Cortex Exhibit Inter-Columnar Synchronization Which Reflects Global Stimulus Properties," *Nature*, *338*:334–337 (March 1989).

33. Grossberg, S. "A neural model of attention, reinforcement and discrimination learning," *7th Hawaii International Conference on System Sciences, p. x+248, 73-4* (1974).

34. Grossberg, S. "Competitive learning: from interactive activation to adaptive resonance," *Cognitive Science, vol.11, no.1, p. 23-63* (1987).

35. Grossberg, S. "3-D vision and figure-ground separation by visual cortex," *Perception and Psychophysics*, 48–120 (1994).

36. Grossberg, S. "Viewnet: a neural architecture for learning to recognize 3D objects from multiple 2D views," *SPIE*, *2353*:266–274 (1994).

37. Grossberg, S. and E. Mingolia. "Neural dynamics of perceptual grouping: Textures, boundaries and emergent segmentation's," *Perception and Psychophysics*, *38*(2):141–171 (1985).

38. Grossberg, Stephen. *Neural Networks and Natural Intelligence*. Cambridge, Massachusetts: MIT Press, 1988.

39. Grossberg, Stephen and Ennio Mingolla. "Neural Dynamics of Motion Perception: Direction Fields, Apertures, and Resonant Grouping," *Perception and Psychophysics*, *53*(3):243–278 (1993).

40. Grossberg, Stephen and Michael E. Rudd. "A Neural Architecture for Visual Motion Perception: Group and Element Apparent Motion," *Neural Networks*, *2*:421–450 (1989).

41. Johnson, J. L.; Ritter, D. "Observation of periodic waves in a pulse-coupled neural network," *Optics Letters, vol.18, no.15, p. 1253-5* (1993).

42. Johnson, J. L. "The pulse coupled neural network as a computer," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2824, p. 66-9* (1996).

43. Johnson, J. L. "Review of products in PCNN [pulse coupled neural network]," *Proceedings of the SPIE - The International Society for Optical Engineering, vol.2878, p. 70-3* (1996).

44. Johnson, J. L. "Weighted products in neural networks," *Proceedings of the SPIE - The International Society for Optical Engineering, vol.2878, p. 74-80* (1996).

45. Johnson, John L. "Pulse-coupled neural nets: Translation, rotation, scale, distortion, and intensity signal invariance for images," *Applied Optics*, *33*(26):6239–6253 (1994).

46. Johnson, John L. "Pulse Coupled Neural Networks," *Proceedings, SPIE Int. Symp. on Opt. Eng. and Photonics in Aerospace Sensing, Conf. Adaptive Computing: Mathematics, Electronics, and Optics (Critical Reviews), Orlando, FL.* (1994).

47. Johnson, John L. "Time Signatures of Images," *Proceedings, IEEE Int. Conf. on Neural Networks, Orlando, FL.* (1994).

48. Jones, J. P. and L. A. Palmer. "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *J. Neurophys.*, *58*:1233–1258 (1987).

49. Kim, Tae-Yong, et al. "KLT-based adaptive vector quantization using PCNN," *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929), p. 4 vol. 3234, 82-7 vol.1* (1996).

50. Kinser, J. and H. J. Caulfield. "An O(N/sup O/) pulse-coupled neural network performing human-like logic," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2760, p. 555-62* (1996).

51. Kinser, J. M.; Johnson, J. L. "Stabilized input with a feedback pulse-coupled neural network," *Optical Engineering, vol.35, no.8, p. 2158-61* (1996).

52. Kinser, J. M. "Object isolation using a pulse-coupled neural network," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2824, p. 70-6* (1996).

53. Kinser, J. M. "A simplified pulse-coupled neural network," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2760, p. 563-7* (1996).

54. Kinser, J. M. "Syntactical computing using pulse-coupled neural network modules," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2824, p. 77-83* (1996).

55. Kinser, J. M. and H. J. Caulfield. "Bioneurological systems," *Proceedings of the SPIE - The International Society for Optical Engineering, vol.2878, p. 60-9* (1996).

56. Kinser, J. M., et al. "Synergistic pulse coupled neural network pattern recognition," *Optical Memory & Neural Networks, vol.5, no.3, p. 179-83* (1996).

57. Konen, W. K., et al. "Fast dynamic link matching algorithm for invariant pattern recognition," *Neural Networks*, *7*(6/7):1019–1030 (1994).

58. Laine, Andrew and Sergio Schuler. "Hexagonal Wavelet Processing of Digital Mammography," *SPIE Image Processing*, *1898* (1993).

59. Laine, Andrew, et al. "Mammographic Feature Enhancement by Multiscale Analysis," *IEEE Transactions on Medical Imaging*, *13*(4) (December 1994).

60. Lawton, T. B. "Dynamic object-based 3-D scene analysis using multiple cues.," *SPIE*, *2054*:194–210 (1984).

61. Lohmann, H., et al. "Visual receptive fields of local intracortical potentials," *Journal of Neuroscience Methods, vol.25, no.1, p. 29-44* (Aug. 1988).

62. Maffei, L. and A. Fiorentini. "The visual cortex as a spatial frequency analyzer.," *Vision Research*, *13*:1255–1267 (1973).

63. Marčelja, S. "Mathematical description of the responses of simple cortical cells," *J. Opt. Soc. Am.*, *70*:1297–1300 (1980).

64. Maunsell, John H. R. "The Brain's Visual World: Representation of Visual Targets in Cerebral Cortex," *Science*, *270*:764–768 (1995).

65. McCandless, Dru, et al. "Model Based Detection of Clustered Microcalcifications using Wavelets and Phantom Derived Images," *Submitted to IEEE Transactions on Medical Imaging* (1996).

66. Mead, C. A. and M. A. Mahowald. "A silicon model of early visual processing," *Neural Networks*, *1*:91–97 (1988).

67. Neven, H. and A. Aertsen. "Rate coherence and event coherence in the visual cortex: a neuronal model of object recognition," *Biological Cybernetics*, *67*:309–322 (1992).

68. Nishikawa, R. M., et al. "Computer-aided detection and diagnosis of masses and clustered microcalcification from digital mammograms," *SPIE*, *1905*:422–432 (1994).

69. Pearlman, William A. "A visual system model and a new distortion measure in the context of image processing," *J. Opt. Soc. Am.*, *68*(3):374–386 (March 1978).

70. Petrovic, Z. R., et al. "Integration of neural networks and genetic algorithms: an example of machine noise diagnosis," *The First World Congress on Intelligent Manufacturing Processes and Systems. Proceedings, p. 2 vol. (xii+xx+1399), 962-71 vol.2* (1995).

71. Polakowski, William, et al. "Computer Aided Breast Cancer Detection and Diagnosis of Masses Using Difference of Gaussians and Derivative Based Feature Saliency," *Submitted to IEEE Transactions on Medical Imaging* (1996).

72. Pollen, D. A., et al. "How does the striate cortex begin the reconstruction of the visual world?," *Science*, *173*:74–77 (1971).

73. Ranganath, H. S. and G. Kuntimad. "Iterative segmentation using pulse coupled neural networks," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2760, p. 543-54* (1996).

74. Ranganath, H. S., et al. "Pulse Coupled Neural Networks for Image Processing." prepublished paper, 1995.

75. Reitboeck, H. J. and J. Altmann. "A model for size- and rotation-invariant pattern processing in the visual system," *Biological Cybernetics, vol.51, no.2, p. 113-21* (1984).

76. Reitboeck, H. J., et al. "Object separation in dynamic neural networks," *1993 IEEE International Conference on Neural Networks (Cat. No.93CH3274-8), p. 3 vol. xxiv+1983, 638-41 vol.2* (1993).

77. Rogers and Kabrisky. *An Introduction to Biological and Artifical Neural Networks for Pattern Recognition*. SPIE Press, 1991.

78. Rogers, S. K., et al. "Neural Networks for Automatic Target Recognition," *IEEE Transactions on Neural Networks*, *8*(7/8):1153–1184 (1995).

79. Ruck, D. W., et al. "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, *1*(4):296–8 (1990).

80. Rumelhart and McClelland. *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. MIT Press, 1986.

81. Sato, Y. and T. Ochiai. "2D genetic algorithms for determining neural network structure and weights," *Evolutionary Programming IV. Proceedings of the Fourth Annual Conference on Evolutionary Programming, p. xx+805, 789-804* (1995).

82. Schneider, J., et al. "Evaluation of neuronal coupling dynamics," *Biological Cybernetics, vol.46, no.2, p. 129-34* (1983).

83. Simmons, P., et al. "Hearing shapes: auditory recognition of two-dimensional spatial patterns," *Proceedings of the International Society for Optical Engineering* (1996).

84. Stoecker, M., et al. "A neural network for scene segmentation by temporal coding," *Neurocomputing, vol.11, no.2-4, p. 123-34* (1996).

85. Strickland, R. N. and H. I. Hahn. "Wavelet Transforms for Detecting Microcalcifications in Mammograms," *IEEE Transactions on Medical Imaging*, *15*(2):218–229 (1995).

86. Swanson, David E. *Retinal Modeling: Segmenting Motion from Spatio-Temporal Inputs Using Neural Networks*. MS thesis, Air Force Institute of Technology, 1992.

87. Syrjakow, M., et al. "Acceleration of direct model optimization methods by function approximation," *Simulation in Industry. 8th European Simulation Symposium. ESS'96, p. 2 vol. (xxxix+666+xv+412), 181-6 vol.2* (1996).

88. Tootel, R. B., et al. "Spatial frequency columns in primary visual cortex.," *Science*, *214*:813–815 (1981).

89. Turner, M. R. "Texture discrimination by Gabor functions," *Biological Cybernetics*, *55*:71–82 (1986).

90. Van Essen, D. C., et al. "Information processing in the primate visual system: an integrated systems perspective," *Science*, *255*:419–422 (1992).

91. von der Malsburg, C. and W. Schneider. "A neural cocktail-party processor," *Biological Cybernetics, vol.54, no.1, p. 29-40* (1986).

92. Wallace, M. T., et al. "Integration of Multiple Sensory Modalities in Cat Cortex," *Experimental Brain Research, 91*(3):484–488 (1992).

93. Werbos, P. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," *PhD dissertation, Harvard University* (1974).

94. Werner, G., et al. "Construction of concepts by the nervous system: from neurons to cognition," *Behavioral Science, vol.38, no.2, p. 114-23* (1993).

95. Wilson, T. A., et al. "Fusion of focus of attention alternatives for FLIR imagery," *Proc. SPIE - Int. Soc. Opt. Eng. (USA), Proceedings of the SPIE - The International Society for Optical Engineering, vol.2756, p. 76-86* (1996).

96. Young, M. P. "The large scale organization of the primate cortical visual system," *SPIE, 2054*:185–193 (1994).

97. Zeki, S. *A Vision of the Brain.* Blackwell Scientific Publications, 1993.

98. Zeki, S. "The visual image in mind and brain." *Mind and Brain* 27–39, W. H. Freeman and Company, 1993.

Vita

Captain Randy P. Broussard [redacted]. He graduated from Crowley High School in 1979. He enlisted in the U.S. Navy in 1980 and spent two years on board the destroyer, U.S.S. William C. Lawe. In 1982 he was awarded a Navy ROTC scholarship, and was released by the Navy with an Honorable Discharge. He entered Tulane University in New Orleans, Louisiana in 1982. In 1983, he was awarded an Air Force ROTC scholarship and transferred to Air Force ROTC. In 1986 he received a Bachelor of Science in Electrical Engineering. Upon graduation he was commissioned as a Second Lieutenant in the U.S. Air Force. His first assignment was at Patrick Air Force Base, Florida, where he served as a computer systems engineer. In 1989 he attended Florida Institute of Technology where he received a Masters of Science degree in Computer Engineering in 1991. In 1991 he was promoted to the rank of Captain and transferred to Los Angeles Air Force Station, California. There he was assigned to the Global Positioning System (GPS) Program office where he served as Computer Directorate Branch Chief. He later served as the GPS Control Systems Engineering Branch Chief. In 1994 he was selected to attend the Air Force Institute of Technology (AFIT) PhD program in residence. His doctoral research focus is physiologically based vision modeling using pulse coupled neural networks.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 26, 1997	Final

4. TITLE AND SUBTITLE
Physiologically-Based Vision Modeling Applications and Gradient Descent-Based Parameter Adaptation of Pulse Coupled Neural Networks

5. FUNDING NUMBERS

6. AUTHOR(S)
Randy Paul Broussard

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Wright-Patterson Air Force Base
2950 P Street
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
PL/LIMI
Phillips Laboratory
Kirtland AFB, NM 87117

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Distribution unlimited. Available for public release.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Pulse coupled neural networks (PCNN) are analyzed and evaluated for use in primate vision modeling, and an adaptive PCNN is developed that automatically sets near-optimal parameter values to achieve a desired output. Biological vision processing principles, such as spatial frequency filtering, competitive feature selection, multiple processing paths, and state dependent modulation are analyzed and implemented to create a PCNN based feature extraction network. This network extracts luminance, orientation, pitch, wavelength, and motion, and can be cascaded to extract texture, acceleration and other higher order visual features. Cortical information linking schemes, such as state dependent modulation and temporal synchronization, are used to develop a PCNN-based visual information fusion network which is used to fuse the results of several object detection systems. Next, the first fully adaptive PCNN is developed. Given only an input and a desired output, the adaptive PCNN finds all parameter values necessary to approximate the desired output. Gradient descent is applied to the PCNN to derive parameter adaptation equations (training rules) for all parameters. Implementing these equations forms a fully adaptive PCNN that minimizes squared error between the actual and desired output. All equations can be applied external to an existing PCNN.

14. SUBJECT TERMS
PCNN, Neural Networks, Automatic Target Detection, Image Fusion, Pulse Coupled, Breast Cancer.

15. NUMBER OF PAGES
114

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unl